DOCUMENT RESUME

ED 240 973 IR 010 646

AUTHOR Johnson, Charles C.; And Others
TITLE BASIC Language Flow Charting Program (BASCHART). Technical Note 3-82.
INSTITUTION Naval Training Analysis and Evaluation Group, Orlando, Fla.
PUB DATE Nov 82
NOTE 74p.; Focus on the Trained Person Series.
PUB TYPE Guides - Non-Classroom Use (055) -- Audiovisual Materials (100) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS *Automation; Computer Oriented Programs; *Computer Programs; *Flow Charts; Guidelines; Program Descriptions; *Programing; *Programing Languages
IDENTIFIERS *BASIC Programing Language

ABSTRACT

This document describes BASCHART, a computer aid designed to decipher and automatically flow chart computer program logic; it also provides the computer code necessary for this process. Developed to reduce the labor intensive manual process of producing a flow chart for an undocumented or inadequately documented program, BASCHART will automatically produce an annotated flow chart for any program that uses WANG (MVP) BASIC-2 programming language. The development of the BASCHART program is summarized, and the hardware requirements are described. A user's guide includes operating procedures and an example of a program flow chart. Three appendices provide (1) a system diagram showing the interrelationships between the program subsystems and a listing of the subroutines with brief descriptions of their purpose; (2) definitions of the BASCHART alphanumeric variables; and (3) the WANG BASIC-2 computer code for the flow charting program. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED240973

TAEG

TRAINING ANALYSIS AND EVALUATION GROUP

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

IR

TECHNICAL NOTE 3-82

BASIC Language Flow Charting Program (BASCHART)

NOVEMBER 1982

FOCUS ON THE TR

...SON

IR010246

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED.

Technical Note 3-82

November 1982

BASIC Language Flow Charting Program (BASCHART)

Charles C. Johnson
Thomas O. Peeples
Gary W. Hodak

APPROVED BY:

Alfred F. Smode
Director

Training Analysis and Evaluation Group
Orlando, Florida 32813

TABLE OF CONTENTS

Section		Page
I	INTRODUCTION	3
	Background	3
	Purpose	4
	Organization of This Report	4
II	SYSTEM DESCRIPTION AND USER'S GUIDE	5
	Development History	5
	Hardware Requirements	5
	BASCHART Operating Procedures	5
	BASCHART Program Flow Chart	12
APPENDIX A	System Diagram of BASCHART Program and BASCHART Subroutine Listing	18
APPENDIX B	BASCHART ALPHANUMERIC VARIABLES	22
APPENDIX C	BASCHART Computer Code Listing	26

LIST OF ILLUSTRATIONS

Figure		Page
A-1	Interrelationship of BASCHART Subroutines	19

SECTION I

INTRODUCTION

Proper computer program documentation is essential to the effective use, maintenance, and modification of computer software. The documentation should be comprised of both user and programmer information and be designed to fill the informational needs of the personnel ultimately responsible for using, maintaining or updating the software. The amount and quality of the software documentation produced are usually dependent on two major factors in the program development cycle-- the length of time allotted for development and the number of people assigned to the development effort. Normally, the longer the duration of a project and the larger the software development team, the greater the need for software documentation. Proper documentation (1) provides a means to efficiently monitor and control long-term projects, (2) provides an understandable, transferrable method of communication between the project team members and future users or program modifiers, and (3) reduces the time consuming, labor intensive process of duplicating the development effort if key project members leave or the program is moved to another site and/or the programs require modification or updating. If the software documentation is adequate, only minimal problems arise when it is modified or transferred to new locations. If the formal documentation is inadequate, or nonexistent, the new user is faced with the formidable and tedious process of manually flow charting the program to determine program logic and defining the program variables prior to proceeding with any modifications. One solution to this approach is to develop a computer aid with the capability to accurately and expeditiously decipher the program logic and automatically provide a flow chart.

BACKGROUND

The Chief of Naval Education and Training (CNET) tasked the Training Analysis and Evaluation Group (TAEG) to document selected software components of the ADP portion of the Foreign Military Sales (FMS) financial management system. The specific objectives of the tasking were to:

- describe the structure of the FMS system and its elements, along with the logic for completing summary billing
- provide an operator's guide that would improve understanding of the program logic and procedures necessary to use the system
- document the specific system programming details of the FMS FY 80 costing/billing programs.

At the outset of this effort, little formal documentation was available other than comments contained in the program software concerning program structure and use. In order to understand how the programs worked, it was necessary to determine program logic and define software variables and their interrelationships. The need to accomplish this formidable labor intensive effort as the necessary prelude to achieving the FMS study objectives served as the impetus for automating this process.

PURPOSE

This document describes a computer aid designed to decipher and automatically flow chart computer program logic and provides the computer code necessary for this process. This computer aid reduces the labor intensive manual process of producing a flow chart for an undocumented or inadequately documented program. This TAEG developed program (hereafter called BASCHART) automatically produces an annotated flow chart for any program that uses WANG (MVP) BASIC-2 programming language.

ORGANIZATION OF THIS REPORT

In addition to this introduction, one other section and three appendices are provided. Section II describes the BASCHART flow charting program and its development and provides a detailed user's guide. Appendix A provides a system diagram of the BASCHART program showing the interrelationships between the subsystems contained in the program. It also contains a listing of the subroutines with brief descriptions of their purpose. Appendix B contains definitions of the BASCHART alphanumeric variables. Appendix C provides the WANG BASIC-2 computer code for the flow charting program.

SECTION II

SYSTEM DESCRIPTION AND USER'S GUIDE

This section briefly summarizes the development of the BASCHART program, describes the hardware requirements, and contains a user's guide for the BASCHART.

DEVELOPMENT HISTORY

The WANG MVP flow charting program is an extensive modification of an existing program developed at the University of Central Florida, Orlando. In the conversion, many of the unique WANG functions were implemented to provide as complete a documentation system as possible. The original program, operated in a batch environment, was designed to work in Harris BASIC on relatively short and noncomplex programs. BASCHART is an online interactive program that can be used on any WANG BASIC-2 program regardless of program length or programmers' writing technique. The program can also isolate a given segment of a program and flow chart only that portion. The user is provided substantial flexibility because BASCHART has been modified to interact with the WANG operating system and disk catalog structure.

HARDWARE REQUIREMENTS

BASCHART is written using WANG BASIC-2 and can be operated on a WANG 2200 MVP within a 56K partition. The user should have a line printer available and must supply the BASCHART program with the program to be flow charted from either a floppy disk or a fixed disk storage medium.

BASCHART OPERATING PROCEDURES

It is assumed that the required computer hardware (CRT, DISK DRIVE, and LINE PRINTER) is available to the user intending to use the BASCHART program. Initializing the equipment is an extremely easy task. However, because of the many possible equipment configurations, it is desirable that personnel knowledgeable in WANG equipment set up the system for subsequent use. After loading and starting the execution of program BASCHART, the following screen will appear.

```
BBBBBB     A     SSSSSS   CCCCCC   H     H     A     RRRRRR   TTTTTTT
B    B    A A    S        C        H     H    A A    R    R      T
BBBBBB   A   A   SSSSSS   C        HHHHHHH   A   A   RRRRRR      T
B    B   AAAAAAA       S   C        H     H   AAAAAAA  R  R        T
B    B   A     A      S    C        H     H   A     A  R   R       T
BBBBBB   A     A  SSSSSS   CCCCCC   H     H   A     A  R    R      T

This program is designed to create a logic flow chart of any active
program on disk in the BASIC-2 language.
Is it your intention to run the program 'BASCHART'?    (Y/N)
```

Entering an N will return the user to the WANG operating system. If the user enters a Y, the following screen will be displayed:

```
          *****  DISK ADDRESS YOUR PROGRAM  *****

     DISK ADDRESSES AVAILABLE:  325  825  365  D10  D11  D13

                    ENTER DISK ADDRESS _ _ _
```

The user must now enter the disk address where the program to be flowcharted is located. Entering a valid disk address will enable the program to scan the specified disk directory and automatically present the following display:

```
                    DISK D11 CATALOG OF PROGRAMS

          10.        20.        30.        40.
 1.       11.        21.        31.        41.
 2.       12.        22.        32.        42.
 3.       13.        23.        33.        43.
 4.       14.        24.        34.        44.
 5.       15.        25.        35.        45.
 6.       16.        26.        36.        46.
 7.       17.        27.        37.        47. BASCHART
 8.       18.        28.        38.        48. BASICout
 9.       19.        29.        39.        49. BASchart

FN '4 = END PROGRAM / FN '5 = SEARCH / FN '6 = BASCHART PROGRAM

                510 TOTAL PROGRAMS     PRESS RTN/FN?
```

This display lists the programs contained at the disk address previously specified. This initial display contains the first 49 programs, after that the programs are presented in groups of 50. Pressing RETURN will cause the next group of 50 programs to appear. Repeating this process will present succeeding groups of programs until all the programs available at that disk address have been displayed. The total number of programs available on the disk is continuously displayed at the bottom of the screen.

Selecting FN '4 = END PROGRAM will return the user to the WANG operating system.

If the user does not know the entire name of the program or its correct spelling, use the search capability of BASCHART, special function Key '5. Selecting FN '5 = SEARCH will cause the screen to display the following:

```
      *****  DISK  D11  CATALOG OF PROGRAMS  *****

      10.          20.          30.          40.
1.    11.          21.          31.          41.
2.    12.          22.          32.          42.
3.    13.          23.          33.          43.
4.    14.          24.          34.          44.
5.    15.          25.          35.          45.
6.    16.          26.          36.          46.
7.    17.          27.          37.          47. BASCHART
8.    18.          28.          38.          48. BASICout
9.    19.          29.          39.          49. BASchart

      INPUT CHARACTER STRING: B A S
```

The user is required to input a character string that approximates the program name. Using this character string, a search of the existing catalog index will be made and all the programs contained on the disk with that character string will be displayed as indicated below:

```
            *****  DISK  D11  SEARCH  *****

47. BASCHART
48. BASICout
49. BASchart

        Three files were searched out with string BAS.

FN '4 = END PROGRAM / FN '5 = SEARCH / FN '6 = BASCHART PROGRAM

            510 TOTAL PROGRAMS     PRESS RTN/FN?
```

The user should now be able to identify the specific program to be flow charted.

Once the user has identified the desired program (either by entering program name or by using the search function, FN '5) and located the program number, special function key '6, FN '6 = BASCHART PROGRAM, is used to initiate the subroutine by which the number of the program (in this instance 47) to be flow charted is entered.

Pressing FN '6 will cause one of the following screens to be displayed. If the user has previously employed the search function (FN '5), the following will appear:

```
                ***** DISK D11 SEARCH *****

47. BASCHART
48. BASICout
49. BASchart

        Three files were searched out with string BAS.

            WHAT IS THE NUMBER OF THE PROGRAM? ____
```

If the user knew the program name and consequently pressed FN '6, the following will be displayed:

```
          ***** DISK D11 CATALOG OF PROGRAMS *****

          10.       20.       30.       40.
1.        11.       21.       31.       41.
2.        12.       22.       32.       42.
3.        13.       23.       33.       43.
4.        14.       24.       34.       44.
5.        15.       25.       35.       45.
6.        16.       26.       36.       46.
7.        17.       27.       37.       47. BASCHART
8.        18.       28.       38.       48. BASICout
9.        19.       29.       39.       49. BASchart

            WHAT IS THE NUMBER OF THE PROGRAM?___
```

Once the user enters a valid program number by either alternative, he has identified the specific program to be flow charted utilizing the BASCHART program.

After entry of a valid program number, a sequence of questions is asked pertaining to printer address, program line number (starting and stopping for the flow chart), and program subroutine structure. The first screen requests a valid printer address.

```
PRINTER ADDRESSES AVAILABLE:  204, 215, 216, 211, 005

SELECT PRINTER ADDRESS:___
```

Entering the desired printer address will cause the following screen to appear:

```
PRINTER ADDRESSES AVAILABLE:  204, 215, 216, 211, 005

SELECT PRINTER ADDRESS:  215

DO YOU WANT A HARD COPY PRINTOUT OF YOUR ENTIRE PROGRAM?  (Y/N) __
```

If a Y is entered, the entire program will be flow charted and printed. If an N is entered, which indicates a desire to print only a portion of the flow charted program, the following display will appear:

```
At what line No. do you want a hard copy print to start?
If first line No. of program, type in '0000'==>  _ _ _ _

At what line No. do you want the hard copy print to end?
If last line No. of program, type in '9999'==>


(Processing is done from the start regardless of where your line No. begins)
```

The user must now enter the starting and stopping line numbers of the portion of the program to be flow charted.

NOTE: If printer address 005 (CRT) is selected, questions pertaining to starting and ending line numbers will not appear. The entire flow charted program will appear on the screen in a continuous rollup display.

Once the program line numbers are entered, the following display, requesting the inclusion of remarks and image statements, appears:

```
DO YOU WISH TO INCLUDE THE REMARKS AND IMAGE STATEMENTS IN THE FLOW CHART?

                              (Y/N)
```

A Y response will include all program remarks and image statements in the flow chart. An N response will exclude all statements of that type and results in a flow chart containing only program logic.

Regardless of which entry is made, the following screen will appear:

```
If your program has marked subroutines, are all of them after the
main program?


Y =  Will cause the oversized heading, 'SUBROUTINES', to be printed
     upon encountering the first marked subroutine.  Everything after
     this is assumed by the program to be subroutine connected.

N =  Will treat all subroutines as if they are found within the main
     program.  No oversized 'SUBROUTINES' heading is printed.




                        SELECT OPTION (Y/N) __
```

A Y response will cause the BASCHART program to print the heading "SUBROUTINES" when it encounters the first subroutine in the program being flow charted. This signifies that the main program is complete and everything that follows the SUBROUTINE heading is, in fact, a subroutine. If the user inserts an N, no heading will be printed and the BASCHART program will treat the subroutines as if they are contained in the main program.

Finally, a list of all the user's responses is presented providing the ability to re-input the data if erroneous results were entered.

```
Your program name is ..............................  BASCHART
Your program is on disk address ..................        D11
Your program will be flow charted on .............        215
You want a printout of the entire program ........         NO
      Starting at line number .....    0
      Ending at line number .......  100
You want remarks & image statements included .....        YES
Your marked subroutines follow the main program...        YES

   Key return to continue, any other key to re-input data  ==> ?
```

After the user enters the desired response, BASCHART will proceed to flow chart the requested program. If a portion of a program is to be flow charted on a printer, and that portion does not begin at zero, the screen will display the flow chart of the program beginning at the first line number. The reason processing is done from the first Line Number, regardless of where the user would like the flow chart to begin, is simply to know if printing is started in a loop or branching sequence. When the starting Line Number to be printed is encountered, the screen displays the status, while flow charting is done at the printer.

During the flow charting process, the BASCHART program will keep the user informed as to status by displaying the following screen:

```
<<<<<<<<<<<<<<  NOW READING  >>>>>>>>>>>>>>

          0190 IF D7$="005"   THEN PRINT HEX(03)

   now at sector #  36086          numbered statements in sector:  1
   Ending sector #  36219        Percentage of processing completed:  2%

.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
..       NOW FORMING A FLOW CHART OF BASCHART ON PRINTER 215          ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..

                    [ REMARKS ARE BEING INCLUDED ]

        OPERATOR NOTE:  PROCESSING TIME VARIES WITH EACH PROGRAM,
                        THE AVERAGE IS 10 MINUTES FOR EVERY 30 SECTORS.
```

The status screen displays the line that is currently being flow charted, the ending sector, the sector being flow charted, and the percentage of processing completed. Once finished, the program returns to query the user as to any more flow chart requirements at this time.

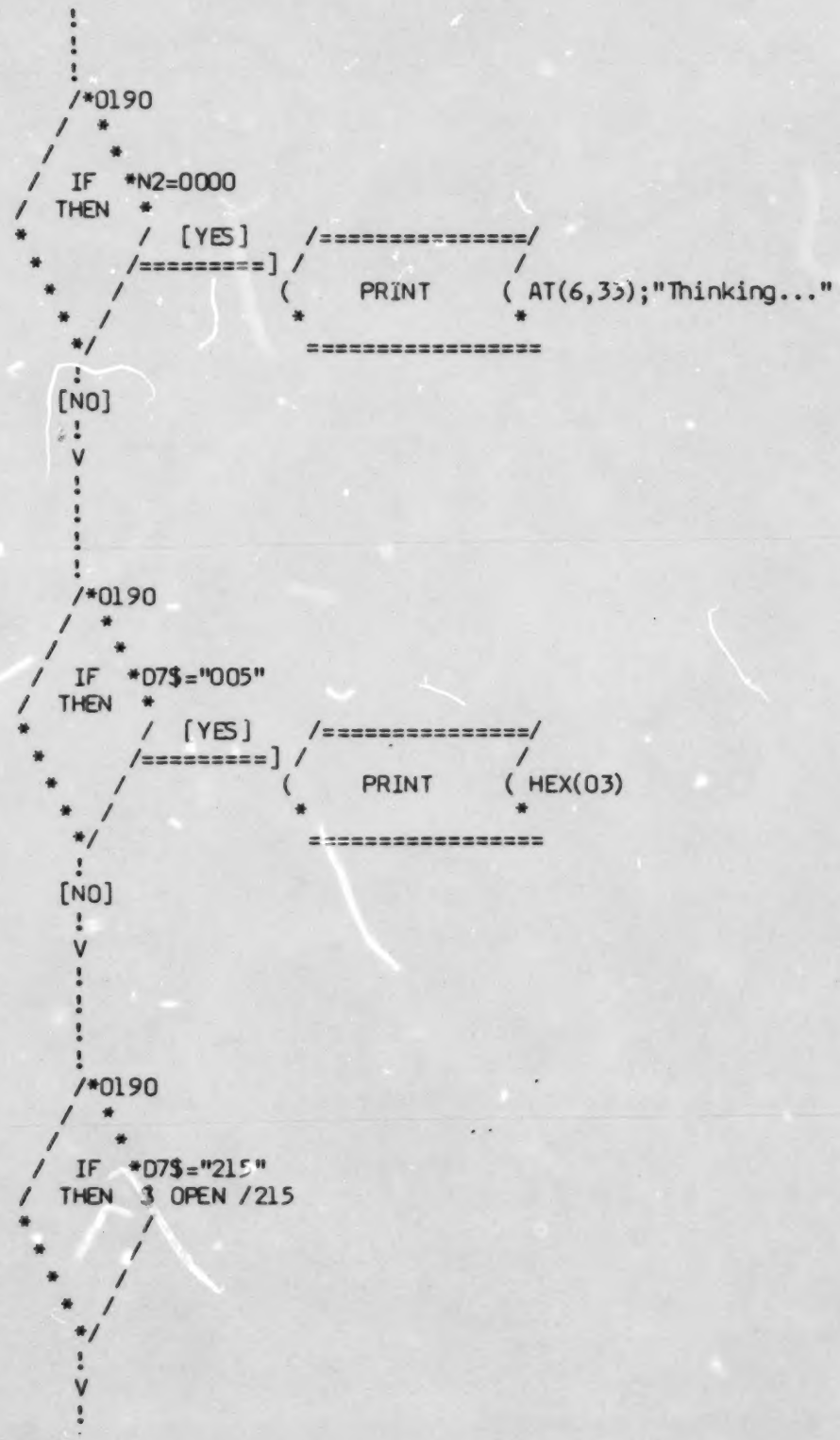
BASCHART PROGRAM FLOW CHART

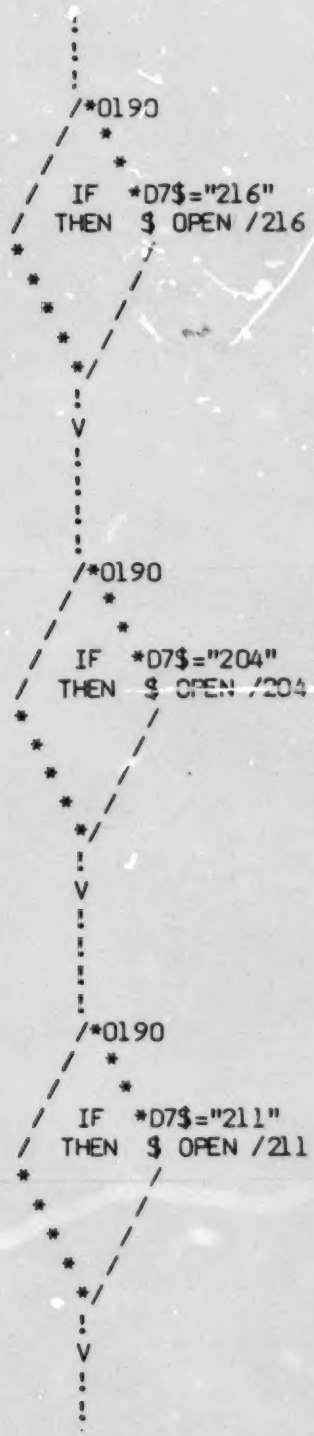
This subsection contains an example of the automated flow charting output produced by the BASCHART program. Program comments are supplied by BASCHART during the processing and at the end of an entire printout. The comments can range from remarks about the programming technique to remarks about a command or commands that could not be translated properly from hexidecimal or exceeded the parameters of the BASCHART program.

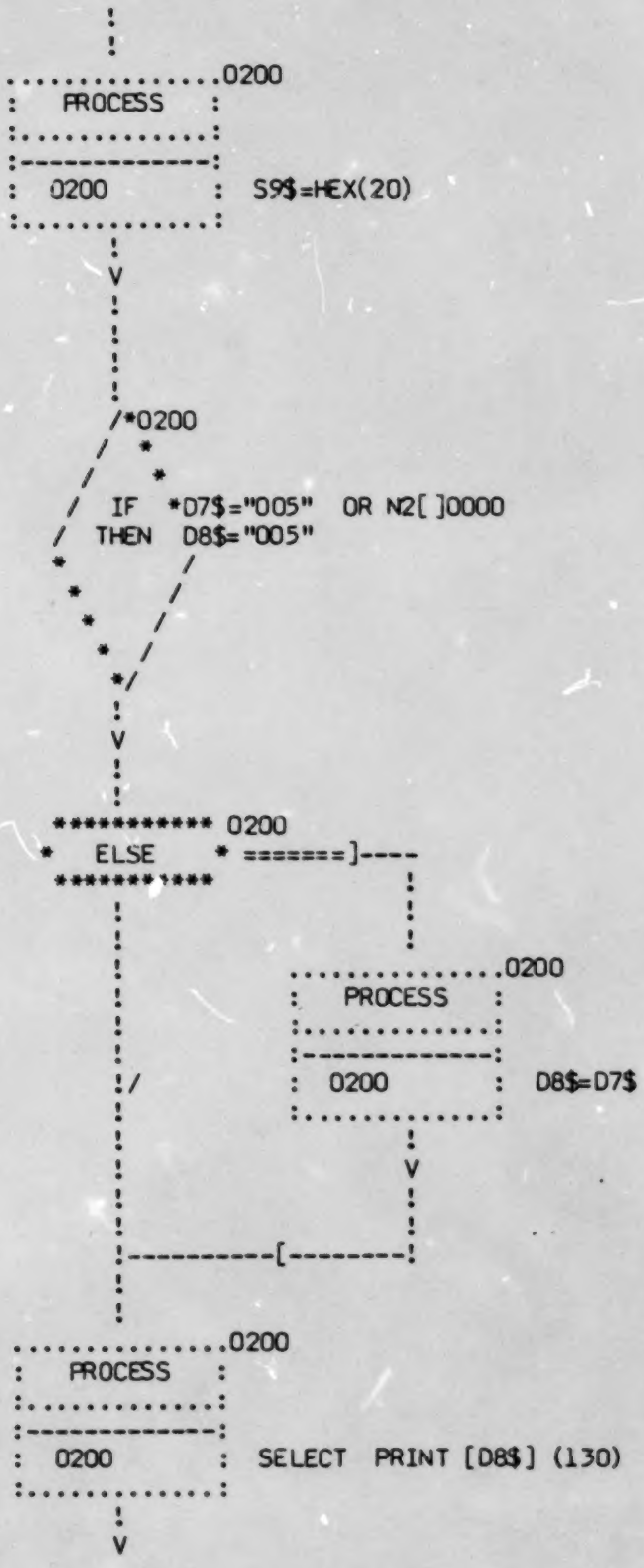
Flow chart of program BASCHART

```
 (---------)
(   START   )
 (---------)
      !
      V
      !
      !
      !
      !
REM %! 'BASCHART' makes a flowchart of any given program
REM ! ( Charles C. Johnson ) TAEG - U.S. NAVY Software
REM                              N.T.C Orlando, Fl.
REM % dimensioned for sector reading
      !
      !
 ..............0120
 :   PROCESS   :
 :.............:
 :-------------:
 :  0120       :  COM Z1
 :-------------:
 :  0120       :  DIM Z$(256)4,Q3$(256)2,Q2$(256)2,Q1$2,Q0$2,X$(256),T$8
 :.............:
      !
      V
      !
      !
REM % dimensioned for the statement conversion
      !
      !
 ..............0120
 :   PROCESS   :
 :.............:
 :-------------:
 :  0120       :  DIM Q7$(123)2,Q6$124
 :.............:
      !
      V
      !
      !
REM % dimensioned for sorting + statement assignment
      !
      !
 ..............0130
 :   PROCESS   :
 :.............:
 :-------------:
 :  0130       :  DIM Q2(256),W$(256)2,L$(256)2,A$124,A1$124
 :.............:
      !
      V
```

```
        :
        :
REM
% :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: ::
% ::          NOW FORMING A FLOW-CHART OF ######## ON PRINTER ###              ::
% :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: ::
        :
        :
  ..............0170
  :   PROCESS   :
  :.............:
  :-------------:
  :  0170       :  SELECT  PRINT 005(80)
  :.............:
        :
        V
        :
        :
        :
        :
   /===============/ 0170
  /  -----------  /
 (   ( PRINTING ( (
  *  -----------  *
   *---------------*
    )---------------)
   /               /
  /    PRINT      /  HEX(0F);HEX(03)
 /===============/
        :
        V
        :
        :
REM % dimensioning for main program
        :
        :
  ..............0180
  :   PROCESS   :
  :.............:
  :-------------:
  :  0180       :  DIM A2$124,B2$124,C2$124,D2$124,F$124,F2$124,K2$124,L3$124,S9$124
  :-------------:
  :  0180       :  DIM E$10,N$5,U1$3
  :-------------:
  :  0180       :  DIM L1$(50)124,L2$(50)124
  :-------------:
  :  0180       :  DIM F0(30),V(30)
  :-------------:
  :  0180       :  G1=0...
  :.............:
        :
        V
        :
```


/*0190
IF *N2=0000
THEN
[YES]
PRINT (AT(6,33);"Thinking..."
[NO]
/*0190
IF *07$="005"
THEN
[YES]
PRINT (HEX(03)
[NO]
/*0190
IF *07$="215"
THEN 3 OPEN /215


/*0190
IF *D7$="216"
THEN $ OPEN /216
*/
/*0190
IF *D7$="204"
THEN $ OPEN /204
*/
/*0190
IF *D7$="211"
THEN $ OPEN /211
*/


0200
PROCESS
0200
S9$=HEX(20)
*0200
IF *D7$="005" OR N2[ ]0000
THEN D8$="005"
0200
ELSE
0200
PROCESS
0200
D8$=D7$
0200
PROCESS
0200
SELECT PRINT [D8$] (130)

APPENDIX A

SYSTEM DIAGRAM OF BASCHART PROGRAM AND BASCHART SUBROUTINE LISTING

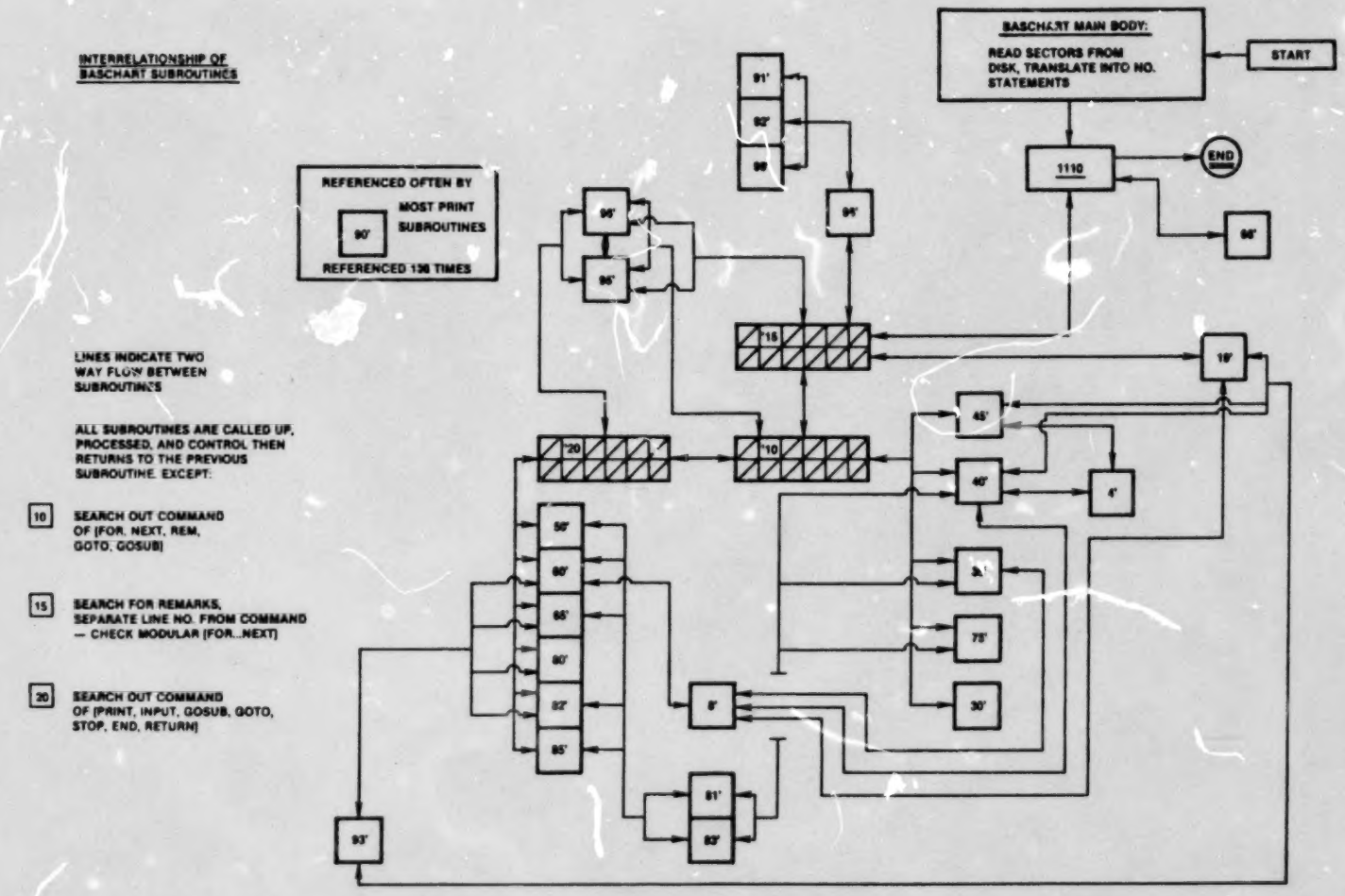

INTERRELATIONSHIP OF BASCHART SUBROUTINES
REFERENCED OFTEN BY
MOST PRINT SUBROUTINES
REFERENCED 136 TIMES
LINES INDICATE TWO WAY FLOW BETWEEN SUBROUTINES
ALL SUBROUTINES ARE CALLED UP, PROCESSED, AND CONTROL THEN RETURNS TO THE PREVIOUS SUBROUTINE. EXCEPT:
10 SEARCH OUT COMMAND OF (FOR, NEXT, REM, GOTO, GOSUB)
15 SEARCH FOR REMARKS, SEPARATE LINE NO. FROM COMMAND — CHECK MODULAR (FOR...NEXT)
20 SEARCH OUT COMMAND OF (PRINT, INPUT, GOSUB, GOTO, STOP, END, RETURN)
BASCHART MAIN BODY:
READ SECTORS FROM DISK, TRANSLATE INTO NO. STATEMENTS
START
1110
END

Figure A-1. Interrelationship of BASCHART Subroutines

BASCHART SUBROUTINE LISTING

	Subroutine	Line	Purpose
	'4	3680	Print direction lines branching horizontally in and out of the flow of logic
	'8	3650	Print (down) direction lines with arrows, prior to printing a new sequence in the flow chart
*	'10	1670	Search new command string for (ON..GOSUB, ON..GOTO, FOR..NEXT, REM, GOTO, GOSUB), AND send to appropriate subroutine
*	'15	1480	Search command string for (REM, %), separate the line No. from the command, in modular phase check for (FOR..NEXT)
	'19	3800	Modular print routine or comment on (IF..THEN) or (FOR..NEXT) statement
*	'20	1830	Search new command string for (IF..THEN, PRINT, INPUT, STOP, END, LOAD, RETURN) AND send to appropriate subroutine
	'30	2420	Print (FOR symbol) in a (FOR..NEXT) Loop, branch out for start of loop
	'35	2520	Print (NEXT symbol) in a (FOR..NEXT) Loop, branch in for end of loop
	'40	2620	Print routine for (ON..GOTO, ON..GOSUB)
	'45	2850	Print (GOTO symbol)
	'50	2920	Print (STOP, END, LOAD symbols)
	'60	2120	Print routine for (IF..THEN) branching sequence and symbol
	'65	2980	Print routine for ELSE, ERROR) Loop branching sequence, begin Loop
	'66	3960	Print routine for (ELSE, ERROR) Loop branching sequence, terminate Loop

*One of three main directional subroutines that decide, based on the present command string, what print subroutine to call up and process.

BASCHART SUBROUTINE LISTING (continued)

Subroutine	Line	Purpose
'75	3040	Print REMARKS from program being flow charted
'80	3550	Print routine for (data process) symbol, unidentified commands default here
'81	3080	Print routine to end (data process) symbol
'82	3860	Print routine for (PRINT symbol)
'83	3940	Print routine to end (PRINT symbol)
'85	3120	Print (RETURN symbol) and/or comment on, no (STOP) or (END) encountered prior to this subroutine
'90	3180	Print flow chart (down) direction lines or straight or in branched, nested Loops
'91	3230	Print comment on Instruction number out of order
'92	3280	Print comment on Instruction number exceeding the program parameters
'93	3320	Print routine for Loop flow direction arrow ('↑' or '↓') up or down
'95	3370	Search the command string for the Operand, or a second main command statement
'96	3440	Print comment on illegal BASIC-2 Language statement encountered
'98	2020	Loop nest counter for the (FOR..NEXT) sequence, check count is within parameters
'99	3500	Print comment on unmatched Loop complement encountered
1100	1110	Search New command string for the First and Last statement of the program and the First and Last statement of a marked subroutine, print summation comments at end of program run

APPENDIX B

BASCHART ALPHANUMERIC VARIABLES

BASCHART ALPHANUMERIC VARIABLES

A$ holds the command string, after isolating from the disk sector and translating from hexcode to normal text

A1$ holds the current command string line number

A2$ holds the first word located in the current command string

B2$ holds the command string, other than the first isolated word of the string

C$ holds the operator input 'Y' or 'N' on the question of marked sub-routines following the main program body

C2$ holds the operand section of the current processed command string

D2$ holds the second command encountered in a large string command (double command string) (i.e., ELSE, ERROR, IF..THEN)

D7$ holds the printer address

D8$ holds the temporary printer address

D9$ holds the disk address

E$ holds the referenced line numbers in (ON..GOTO), (ON..GOSUB) statements

F$ holds temporary first word of the current command string

F2$ holds the variable and starting value in a (FOR..NEXT) statement, to use later in identifying the end of loop

K2$ holds the (FOR..) or (..NEXT) command in the modular phase of the flow chart being created

L3$ holds the (FOR..) or (..NEXT) command during a check routine to match each with its correct complement

N$ holds the present statement line number

Q0$ holds the locations of the starting line numbered statements, for each sector of the program being processed

Q1$ holds the location of the end of sector data marker (FD) or (FE) for each sector of the program being processed

Q6$ holds the first byte of each command string and, if needed, the translated ASCII of the hexcoded command byte

S9$ holds a blank field, used in translating statements during the flow charting sequence of the program

BASCHART ALPHANUMERIC VARIABLES (continued)

T$ holds the name of the program being flow charted

T1$ holds the temporary value of 'S9$', used as a flag when encountered to indicate some operation was completed

T2$ holds the entire (intact) command string, while elsewhere it is being broken down and processed during the flow chart sequence

U1$ holds the location in a command string, while being read from the sector, of the actual end of the statement, not counting the ':' (colon), which can be a statement separator

Z1$ holds the operator input 'Y' or 'N' on the question of including remarks and image statements in the flow chart output

BASCHART ALPHANUMERIC ARRAY VARIABLES

L$( )	holds the sorted sector location of each beginning string command
L1$( )	holds the stored, nested (FOR..NEXT) statement, to match against each equal complement
L2$( )	holds the stored line number of each nested, starting (FOR..NEXT) statement
Q2$( )	holds locations of each start of numbered statements in each sector, read by the program
Q3$( )	holds locations of each start of nonnumbered statements in each sector, read by the program
Q4$( )	holds the ASCII equivalent of a hexcoded command
Q5$( )	holds the hexcoded commands used in the BASIC-2 Language
Q7$( )	holds the sector position of each ":" mark, and later each "%", in the sector currently being read
W$( )	holds the work array for the MAT SORT command
X$( )	holds the entire sector being read by the program
Z$( )	holds the temporary location of "%" statements in the process of isolating string commands, later it holds the sorted location, in the sector being read, of each string command prior to its being translated from hexcode into ASCII

APPENDIX C

BASCHART COMPUTER CODE LISTING

```
10/25/82      BASchart        PAGE  1
 PROGRAM LISTING

       10 REM %!
10/25/82      BASchart        PAGE  2
 PROGRAM LISTING

         BASchart - start program for

         the BASCHART flowchart: REM ! [ Charles C. Johnson ] TAEG-U.S. NAVY S
         oftware : REM                              N.T.C. Orlando, Florida
        20 REM    C$(450)8  -  File Names from the Disk Directory.
        30 REM    C1$2      -  File Status in hex.
        40 REM ==========================================================
        50 REM %

         dimensioning

        60 COM D9$3,T$8,D8$3,D7$3,Q4$(123)10,Q5$(123)10,Z1$1,C$1,N2,N3
         :DIM N2$4,N3$4,Z2$1,Z3$3,Z4$3,Z5$3,A$3
         :DIM N$8,S$2,S2$2,Z$3,Z1$1
        70 DIM B$(16)16,C$(1100)8,C1$2,C2$(1100)8,L$(1100)2,W$(1100)2: REM %---
         ------------------------------------------------------------------
         -----
         :$PSTAT="BASCHART"
        80 C$()=ALL(FF)
        90 IF A=1  THEN  230
         :IF Z1=1  THEN  210
       100 PRINT HEX(06)
         :FOR G=1  TO 3
         :E=11
         :F=36
         :H=0
       110 H=H+1
         :PRINT HEX(0306)
         :E=E-1
         :F=F-4
         :PRINT HEX(06);AT(E,F);BOX(H*2,H*8)
         :IF E[]0 AND F[]0  THEN  110
         :NEXT G
         :PRINT HEX(03)
       120 %    BBBBBB     A     SSSSSS   CCCCCC   H     H     A     RRRRRR  T
         TTTTTT
       130 %    B    B    A A    S        C        H     H    A A    R    R
          T
       140 %    BBBBBB   A   A   SSSSSS   C        HHHHHHH   A   A   RRRRRR
          T
       150 %    B    B  AAAAAAA       S   C        H     H  AAAAAAA  R  R
          T
       160 %    B    B  A     A       S   C        H     H  A     A  R   R
          T
       170 %    BBBBBB  A     A  SSSSSS   CCCCCC   H     H  A     A  R    R
          T
       180 PRINT AT(7,0);
         :PRINTUSING  120
         :PRINTUSING  130
         :PRINTUSING  140
         :PRINTUSING  150
         :PRINTUSING  160
```

PROGRAM LISTING

```
   :PRINTUSING 170
190 PRINT AT(17,10);"This program is designed to create a logic flowcha
   rt of"
   :PRINT AT(18,12);"any active program on disk in the Basic-2 language.
   "
200 Z1$=HEX(0D)
   :PRINT AT(22,7);"Is it your intention to run the program 'BASCHART' ?
     (Y/N)";
   :KEYIN Z1$
   :IF Z1$="N" OR Z1$="n"  THEN GOSUB '4
   :IF Z1$[ ]"Y" AND Z1$[ ]"y"  THEN  200
   :PRINT HEX(03)
210 A=1
220 REM %

  ========= SELECT DISK

                          ==========
   :GOSUB '205
230 DATA LOAD BA T#1,(0)B$()
   :ERRORPRINT HEX(07)
   :GOTO  220

240 % ************************************************************************
   *********
250 % *              Now Loading All the Active Programs on ###
          *
260 % ************************************************************************
   ********
270 PRINT HEX(0306);AT(8,0);
   :PRINTUSING  240
   :PRINTUSING  250,D9$
   :PRINTUSING  260
   :I=0
280 S2$=STR(B$(1),1,2)
   :S=VAL(STR(S2$,2,1))
290 S$=ALL(FF)
   :L=0
   :L1=0
   :C1$=HEX(1080)
300 PRINT HEX(06);AT(18,25);"LOADING CATALOG";
310 REM %

  =========  LOAD FILE NAMES INT

   0 C$()              ==========
320 S$=ADDC BIN(1)
330 IF S$]=S2$  THEN  420
340 DATA LOAD BA T#1,(S$)B$()
350 PRINT AT(18,45,5);AT(18,45); S;
   :S=S-1
360 FOR J=1  TO 16
```

PROGRAM LISTING

```
370 IF STR(B$(J),1,2)[]STR(C1$,1,2)  THEN  390
380 L=L+1
   :C$(L)=STR(B$(J),9,8)
390 NEXT J
400 GOTO  320

410 REM %

  =========  SORT FILE NAMES

                          ==========
420 PRINT HEX(06);AT(18,25);"    Sorting the Catalog      "
   :IF L=0  THEN  500
   :MAT SORT C$()  TO W$(),L$()
430 MAT MOVE C$(),L$(1)  TO C2$(1)
   :PRINT HEX(0306)
440 FOR L2=1  TO L
445 PRINT HEX(06);AT(0,20);"***** Disk ";D9$;" Catalog of Programs *****
   "
450 IF MOD(L2,10)=0  THEN L1=L1+15
460 IF L1 [66  THEN  480
   :L1=0
470 GOSUB  810
480 PRINT HEX(06);AT(MOD(L2,10)+3,L1);L2;TAB(L1+4);C2$(L2)
490 NEXT L2
500 GOSUB  810
   :L1=0
   :GOTO  440

540 REM %

  =========  DEFFN'5  / SEARCH R

   OUTINE                 ==========
550 DEFFN'5
560 PRINT AT(22,0,79)
570 PRINT AT(22,20);"Input character string:";
   :LINPUT-N$
580 IF N$[ ]"          "  THEN  600
590 PRINT AT(22,0,79);AT(22,0);
   :RETURN

600 MAT SEARCH C2$(),=N$  TO L$()
610 IF L$(1)]HEX(0000)  THEN  640
620 PRINT AT(22,52);"NOT ON FILE"
630 GOTO  570

640 J,I1=1
   :J1=2
   :Y1=0
   :FOR K=0  TO 21
   :PRINT AT(K,0,30)
```

PROGRAM LISTING

```
   :NEXT K
   :K=1
650 IF L$(J)=HEX(0000) THEN 780
655 PRINT HEX(06);AT(0,20);"***** Disk ";D9$;" Search of Programs *****
   "
660 X=VAL(L$(J),2)
670 Y=INT((X-1)/8)+1
680 IF Y1=Y THEN 715
690 I=MOD(J1,10)+1
700 IF I[]1 THEN 720
710 I1=I1+15
   :J1=1
   :I=2
   :IF I1]70 THEN I1=0
   :GOTO 720

715 IF MOD(J,45)[]0 THEN 760
   :PRINT AT(18,12-LEN(N$));K-1;"Files were searched out having ";N$;" (
   more on next screen)"
   :GOSUB 810
   :I=2
   :J1,I1=1
   :GOTO 760

720 IF MOD(J,45)[]0 THEN 730
   :PRINT AT(18,12-LEN(N$));K-1;"Files were searched out having ";N$;" (
   more on next screen)"
   :GOSUB 810
   :I=2
   :J1,I1=1
730 PRINT AT(I+3,I1+4);BOX(1,9);AT(I+3,I1);Y;AT(I+3,I1+5);C2$(Y)
   :K=K+1
740 Y1=Y
750 J1=J1+1
760 J=J+1
770 IF J[=L THEN 650
780 PRINT AT(18,20-LEN(N$));K-1;"Files were searched out with string ";N
   $;
   :GOSUB 810
790 N$=ALL(20)
   :GOTO 570

800 REM %

  ========= LOAD ROUTINE

                          ==========
810 PRINT HEX(06);AT(20,0,79);AT(20,2);"[ FN'4 = End / FN'5 = Search / F
   N'6 = Program you want to run on BASCHART ]"
   :PRINT HEX(06);AT(22,0,79);AT(22,17);L;" Total Programs    ";HEX(06);
   :INPUT "  PRESS [RTN/FN] ",A$
820 IF A$=HEX(0D2020) OR A$=" " OR A$="[R]" THEN 830
```

PROGRAM LISTING

```
   :A$="[R]"
   :GOTO 810

830 PRINT HEX(03)
   :RETURN

840 REM %

  ========  DEFFN'205 / SELECT

   DISK SU  OUTINE    ==========
850 DEFFN
   :PRINT H  );AT(12,11);"Disk Addresses available:  325 B25 365 D10
   D11 D12 D  920";AT(13,38);"D14 D15 D30 D31 D32 D33 D34 D35"; AT(12,8
   );BOX(2,61);AT(9,9);"**** Disk address of the program you want to Flo
   wchart ****"
860  PRINT AT(18,24);"Select Disk Address:  ";
   :D9$=" "
   :LINPUT-D9$
   :PRINT HEX(06)
   :IF D9$=" "  THEN GOSUB '4
   :IF POS("3BD"=STR(D9$,,1))*POS("123456"=STR(D9$,2,1))*POS("012345"=
   STR(D9$,3,1))=0  THEN  860
   :SELECT #1[D9$]
   :RETURN

870 REM %

  ========= DEFFN'            UR

   N START             ==========
880 DEFFN'4
   :$PSTAT=" "
   :COM CLEAR
   :RETURN  CLEAR  ALL

   :LOAD T "START"

   :ERRORPRINT HEX(03);AT(8,19);"NOT ABLE TO LOAD START"
   :END

890 REM %

  ========= DEFFN'6 / LOAD BASCH

   ART                 ==========
900 DEFFN'6
910 PRINT AT(22,0,79)
920 Z=0
   :PRINT AT(22,10);"
            "
   :PRINT AT(22,16);"WHAT IS THE NUMBER OF THE PROGRAM:  ";
```

PROGRAM LISTING

```
    :LINPUTZ$
    :IF Z$=" "  THEN RETURN
    :CONVERT Z$  TO Z
    :ERRORGOTO  920
930 Z1$=HEX(0D)
    :IF Z]L OR Z[1  THEN  920
    :PRINT AT(22,14);"
      "
    :PRINT AT(22,16);"IS ";C2$(Z);" THE PROGRAM YOU WANT ? (Y/N) ";
    :KEYIN Z1$
940 IF Z1$="N" OR Z1$="n"  THEN  920
    :IF Z1$[ ]"Y" AND Z1$[ ]"y"  THEN  930
950 DIM D8$(5)3,C8$2
    :PRINT HEX(03);AT(9,13);"Printer Address available:  204 215 216 211
    005"
    :PRINT AT(12,21);"Select Printer Address:  ";
    :D8$=" "
    :LINPUT-D8$
960 STR(D8$(),1)= "204215216211005"
    :MAT SEARCH D8$(),=D8$  TO C8$  STEP 3
    :IF C8$=HEX(0000)  THEN  950
970 DATA "AND","RUN","CLEAR","RENUMBER","TAPE","OR","DISK","TEMP","XOR",
    "KEYIN","COPY"
    :DATA "DSKIP","LIMITS","LIST","SAVE","CONTINUE","NEXT","IF","GOTO","G
    OSUB","RETURN","FOR"
    :DATA "DATA","READ","LET","LINPUT","END","DIM","STOP","TRACE","ON","L
    S","ALL","CLOSE"
980 DATA "DAC","DSC","SUB","ROTATE","PACK","UNPACK","BOOL","ADD","INIT",
    "ERROR","ERR"
    :DATA "VERIFY","DBACKSPACE","BEG","OFF","CI","CO","D","R","OPEN","LIN
    PUT","ELSE"
990 DATA "ROUND","SPACE","PRINTUSING","CONVERT","MOVE","PLOT","PRINT","R
    EM","COM","RESTORE"
    :DATA "SELECT","LOAD","MAT","PLOT","REWIND","BACKSPACE","SKIP","SCRAT
    CH","DA","DC"
    :DATA "BA","THEN","TO","STEP","G","P","BT","STR(","HEX(","RE","%","BI
    N(","LEN("
1000 DATA "VAL(","NUM(","POS(","#","ATN(","SIN(","COS(","ARC(","TAB(","DE
    FFN"
    :DATA "FN","#PI","ABS(","COS(","EXP(","INT(","LOG(","SIN(","SGN(","SQ
    R(","RND(","TAN("
    :DATA "TAN(","FIX(","HEX","$","LGT(","HEX(","AT(","HEXCF(","MAX(","MI
    N(","MOD(","VER("
1010 REM
1020 DATA "8A","82","81","83","8F","88","8E","8D","8C","88","87"
    :DATA "89","86","80","85","84","9D","9F","9C","9A","9B","9E"
1030 DATA "97","98","91","99","96","93","95","90","94","ED","E1"
    :DATA "E3","ED","EE","EF","E9","E2","E6","E7","E8","E4","EB"
1040 DATA "EC","BC","B8","B3","BA","B5","B8","B7","B6","B4","F0"
1050 DATA "F2","F4","F3","A7","AE","AD","AF","A0","A2","A6","A3"
    :DATA "A5","A1","A8","A4","A9","AB","AA","AC","BD","BF","BE"
1060 DATA "B1","B2","B0","DB","D9","DA","D3","D2","D6","D8","DE"
```

PROGRAM LISTING

```
    :DATA "D5","DC","DD","DF","D7","D4","D0","D1","CB","CD","CE"
1070 DATA "C0","CC","C1","C3","C4","C5","C6","C7","C8","C2","C9"
    :DATA "CA","CF","92","E5","EA","B9","FC","F5","F6","F7","F8"
    :DATA "F9","F1"
1080 REM
1090 RESTORE LINE 970
    :FOR I=1 TO 123
    :READ Q4$(I)
    :NEXT I
1100 RESTORE LINE 1020
    :FOR I=1 TO 123
    :READ Q5$(I)
    :NEXT I
1110 Z2$=" "
    :D7$=D8$
    :N2=0000
    :N3=9999
    :IF D8$="005" THEN 1170
    :Z2$=HEX(0D)
1120 PRINT AT(15,6);"Do you want a hard copy print-out of your entire pro
    gram ? (Y/N)";
    :KEYIN Z2$
    :IF Z2$="N" OR Z2$="n" THEN 1130
    :IF Z2$[ ]"Y" AND Z2$[ ]"y" THEN 1120
    :GOTO 1170

1130 PRINT HEX(03);AT(24,3);"(Processing is done from the start regardle
    ss where your line No. begins)"
1140 PRINT AT(10,10);BOX(2,58);AT(10,11);"At what line No. do you want a
    hard copy print to start ? "
    :PRINT AT(11,11);"If first line No. of program, type in '0000'==]";
    :LINPUT-N2$
    :PRINT AT(5,2,78)
    :CONVERT N2$ TO N2
    :ERRORPRINT AT(5,32);".PLEASE TRY AGAIN."
    :GOTO 1140

1150 PRINT AT(14,10);BOX(2,58);AT(14,11);"At what line No. do you want th
    e hard copy print to end ? "
    :PRINT AT(15,11);"If last line No. of program, type in '9999'==]";
    :LINPUT-N3$
    :PRINT AT(5,2,78)
    :CONVERT N3$ TO N3
    :ERRORPRINT AT(5,32);".PLEASE TRY AGAIN."
    :GOTO 1150

1160 IF N3]N2 THEN 1170
    :PRINT AT(5,20);".Sorry, your line No.'s are out of order."
    :GOTO 1140

1170 Z1$=HEX(0D)
    :PRINT HEX(03);AT(5,0);BOX(2,78)
```

PROGRAM LISTING

```
        :PRINT AT(5,2);"Do you wish to include the remarks and image statemen
        ts in the flow-chart?"
        :PRINT AT(6,36);"(Y/N)";
        :KEYIN Z1$
        :IF Z1$="y"  THEN Z1$="Y"
        :IF Z1$="n"  THEN Z1$="N"
        :IF Z1$="N" OR Z1$="Y"  THEN 1180
        :GOTO 1170

    1180 C$=HEX(0D)
        :PRINT AT(10,0);BOX(10,79);AT(10,1);"If your program has marked subro
        utines,are all of them after the main program?"
    1182 PRINT HEX(06);AT(13,5);"Y =] Will cause the oversized heading, 'SUBR
        OUTINES', to be printed";AT(14,10);"upon encountering the first marke
        d subroutine. Everthing after";AT(15,10);"this is assumed by the prog
        ram to be subroutine connected."
    1184 PRINT HEX(06);AT(17,5);"N =] Will treat all subroutines as if they a
        re found within the";AT(18,10);"main program. No oversized 'SUBROUTIN
        ES' heading is printed."
    1185 PRINT AT(11,36);"(Y/N)";
        :KEYIN C$
        :IF C$="y"  THEN C$="Y"
        :IF C$="n"  THEN C$="N"
        :IF C$="N" OR C$="Y"  THEN 1400
        :GOTO 1170

    1190 % :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: ::
        :: :: :: ::
    1200 % ::
                  ::
    1210 % ::
                 ::
    1220 % ::  Your program name is .....................................
        ########  ::
    1230 % ::
                 ::
    1240 % ::  Your program is on disk address ..........................
             ###  ::
    1250 % ::
                 ::
    1260 % ::  Your program will be flow-charted on .....................
             ###  ::
    1270 % ::  Your program will be flow-charted on the .................
         SCREEN  ::
    1280 % ::
                  ::
    1290 % ::  You want a print-out of the entire program ...............
             ###  ::
    1300 % ::
                  ::
    1310 % ::        Starting at line number .....  ####
                 ::
```

PROGRAM LISTING

```
1320 % ::
                ::
1330 % ::          Ending at line number .......  ####
               ::
1340 % ::
                ::
1350 % ::  You want remarks & image statements included .............
           ###  ::
1360 % ::
                ::
1370 % ::  Your marked subroutines follow the main program ..........
           ###  ::
1380 % ::
                ::
1390 % :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: ::
    :: :: :: ::
1400 IF N2=0000 AND N3=9999  THEN Z3$="YES"
    :ELSEZ3$=" NO"
    :IF Z1$="Y"  THEN Z4$="YES"
    :ELSEZ4$=" NO"
    :IF C$="Y"  THEN Z5$="YES"
    :ELSEZ5$=" NO"
    :PRINT HEX(03)
1410 PRINT
    :PRINTUSING 1190
    :PRINTUSING 1200
    :PRINTUSING 1210
    :PRINTUSING 1220,C2$(Z)
    :PRINTUSING 1230
    :PRINTUSING 1240,D9$
    :PRINTUSING 1250
    :IF D7$[ ]"005"  THEN PRINTUSING 1260,D7$
    :ELSEPRINTUSING 1270
    :PRINTUSING 1280
    :PRINTUSING 1290,Z3$
1420 PRINTUSING 1300
    :PRINTUSING 1310,N2
    :PRINTUSING 1320
    :PRINTUSING 1330,N3
    :PRINTUSING 1340
    :PRINTUSING 1350,Z4$
    :PRINTUSING 1360
    :PRINTUSING 1370,Z5$
    :PRINTUSING 1380
    :PRINTUSING 1390
    :PRINT AT(5,65);BOX(1,9)
1430 Z2$=" "
    :PRINT AT(23,6);"Press RETURN to continue, any other key to re-input
    data ==] ? ";
    :KEYIN Z2$
    :IF Z2$=HEX(0D)  THEN 1440
    :Z1=1
```

PROGRAM LISTING

```
          :D9$,D7$,N2$,N3$,Z$,Z1$,C$=HEX(20)
          :A=0
          :GOTO   80

       1440 PRINT HEX(03);HEX(020402020E);AT(8,18);"Now loading main BASCHART p
          rogram"
          :T$=C2$(Z)
          :RETURN  CLEAR  ALL

          :LOAD T"BASCHART"

       9000 REM %

          GENERAL SUBROUTINE

          :PRINT HEX(030E);"YOU HAVE LOADED 'DEFFNO' THE GENERAL SUBROUTINES";
          HEX(0A)
          :LIST
          :END

       9010 DEFFN 'O "SCRATCHT";HEX(22);"BASchart";HEX(22);":SAVET ()";HEX(22);
          "BASchart";HEX(220D)
       9070 DEFFN '31
          :GOSUB '4
```

```
10/25/82      BASCHART        PAGE  1
 PROGRAM LISTING

      100 REM %:

10/25/82      BASCHART        PAGE  2
 PROGRAM LISTING

         'BASCHART' makes a flowchart

         of any given program
      110 REM ! ( Charles C. Johnson ) TAEG - U.S. NAVY Software: REM
                              N.T.C Orlando, Fl.
      120 REM %

         dimensioned for sector readin

         g
         :COM Z1
         :DIM Z$(256)4,Q3$(256)2,Q2$(256)2,Q1$2,Q0$2,X$(256),T$8: REM % dimens
         ioned for the statement conversion
         :DIM Q7$(123)2,Q6$124
      130 REM %

         dimensioned for sorting + sta

         tement assignment
         :DIM Q2(256),W$(256)2,L$(256)2,A$124,A1$124: REM
      140 % :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: ::
         :: :: ::
      150 % ::          NOW FORMING A FLOW-CHART OF ######## ON PRINTER ###
                ::
      160 % :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: :: ::
         :: :: ::
      170 SELECT PRINT 005(80)
         :PRINT HEX(0F);HEX(03)
      180 REM %

         dimensioning for main program

         :DIM A2$124,B2$124,C2$124,D2$124,F$124,F2$124,K2$124,L3$124,S9$124,T1
         $124,T2$124
         :DIM E$10,N$5,U1$3
         :DIM L1$(50)124,L2$(50)124
         :DIM F0(30),V(30)
         :G1=0
      190 IF N2=0000  THEN PRINT AT(6,33);"Thinking..."
         :IF D7$="005"  THEN PRINT HEX(03)
         :IF D7$="215"  THEN $OPEN /215
         :IF D7$="216"  THEN $OPEN /216
         :IF D7$="204"  THEN $OPEN /204
         :IF D7$="211"  THEN $OPEN /211
      200 S9$=HEX(20)
         :IF D7$="005" OR N2[ ]0000  THEN D8$="005"
         :ELSED8$=D7$
         :SELECT PRINT [D8$] (130)
         :PRINT HEX(0D)
         :PRINT "                                                    Flow chart of pr
         ogram  ";T$
      210 IF D8$[ ]"005"  THEN PRINT "////////////////////////////////////////////
```

10/25/82 BASCHART PAGE 3
PROGRAM LISTING

```
         ////////////////////////////////////////////////////////////////////////////////
         ////////////////"
      220 PRINT "                "
         :PRINT "  (---------)"
         :PRINT " (   START   )"
         :PRINT "  (---------)"
         :PRINT "        !"
         :PRINT "        V"
         :PRINT "        !"
         :PRINT "        !"
         :PRINT "        !"
         :PRINT "        !"
      230 SELECT #1 [D9$]
      240 LIMITS T#1,T$,0,Q0,Q1,Q1
         :Q8=0
      250 IF ABS(Q1)=1  THEN  260
         :SELECT PRINT 005(80)
         :PRINT HEX(03);AT(12,8);"ERROR, ";T$;" is not an active program";AT(1
         4,8);"now Loading BASCHART start program"
         :GOTO  730

      260 Q8=Q8+1
         :DATA LOAD BA T#1,(0+Q8) X$()
         :MAT SEARCH X$(),=HEX(FE)  TO Q1$
         :IF Q8+0=Q0  THEN  270
         :IF VAL(Q1$,2)=0  THEN  260
         :Q0=0+Q8
      270 REM %

         READ SECTORS ON DISK, ONE AT

         A TIME
      280 0=0+1
         :DATA LOAD BA T#1,(0) X$()
      290 Q2$(),Q3$(),Z$(),W$(),L$(),Q7$()=ALL(00)
         :Q1$,Q0$,Q6$=HEX(00)
         :MAT Q2=ZER
         :E5,I=0
      300 MAT RE DIM Q2(256),L$(256)2,W$(256)2
      310 REM %

         SEARCH FOR BEGINNING AND ENDI

         NG LINE STATEMENTS
      320 IF STR(X$(),2,2)=HEX(0000)  THEN G1=1
         :IF G1=1  THEN 1310
      330 MAT SEARCH X$(),=HEX(FD)  TO Q1$
         :IF VAL(Q1$,2)=0  THEN MAT SEARCH X$(),=HEX(FE)  TO Q1$
      340 MAT SEARCH X$(),=HEX(20FF)  TO Q0$
         :MAT SEARCH X$(),=HEX(00FF)  TO Q2$()
      350 MAT SEARCH X$(),=HEX(3A)  TO Q3$()
      360 IF VAL(Q0$,2)]2  THEN Q0$=ALL(00): REM
```

10/25/82 BASCHART PAGE 4
PROGRAM LISTING

```
   ( delete colons in between '"' marks )
370 MAT SEARCH X$(),=HEX(22)  TO Q7$()
   :IF VAL(Q7$(1),2)=0 OR VAL(Q3$(1),2)=0  THEN  420
380 I3,I=1
390 IF Q3$(I)]Q7$(I3) AND Q3$(I)[Q7$(I3+1)  THEN  400
   :IF Q3$(I)[Q7$(I3+1)  THEN I=I+1
   :IF Q3$(I)]Q7$(I3+1)  THEN I3=I3+2
   :IF VAL(Q7$(I3+1),2)=0 OR VAL(Q3$(I),2)=0  THEN  420
   :GOTO  390

400 I2=I
410 Q3$(I2)=Q3$(I2+1)
   :IF VAL(Q3$(I2+1),2)=0  THEN  390
   :I2=I2+1
   :GOTO  410

420 I2,I,I3=0
   :Q7$(),Z$()=ALL(00)
   :IF U5=1  THEN  490
430 REM %

   delete colons in between % an

   d next line No.
   :MAT RE DIM Z$(256)2
   :MAT SEARCH X$(),=HEX(D8)  TO Z$()
   :IF VAL(Z$(1),2)=0  THEN  490
   :U5=1
   :Q7$(1)=Z$(1)
   :I2=1
440 I2=I2+1
   :IF VAL(Z$(I2),2)=0  THEN  450
   :Q7$(I2+I2-1)=Z$(I2)
   :GOTO  440

450 Z$()=ALL(00)
   :I2=0
460 I2=I2+2
   :IF I2=2  THEN I2=1
   :IF VAL(Q7$(I2),2)=0  THEN  380
470 MAT SEARCH STR(X$(),VAL(Q7$(I2),2)),=HEX(0D)  TO U1$
   :I=VAL(U1$,2)+VAL(Q7$(I2),2)+3
480 Q7$(12+1)=BIN(I,2)
   :GOTO  460

490 MAT RE DIM Z$(256)4
   :U5=0: REM % number of statements with line numbers
500 I=I+1
   :IF VAL(Q2$(I),2)[]0  THEN  500
   :IF VAL(Q0$,2)[]0  THEN I=I+1
   :E5=I-1
510 REM %
```

10/25/82 BASCHART PAGE 5
PROGRAM LISTING

```
        CONVERTING Q1$,Q0$,Q2$(,Q3$(

         TO  Z$()
      520 I=1
         :I2,I3,Q=0
      530 Q=VAL(Q1$,2)
         :CONVERT Q  TO Z$(I),(####)
         :IF VAL(Q0$,2)=0  THEN  540
         :G1=1
         :I=2
         :Q=VAL(Q0$,2)+1
         :CONVERT Q  TO Z$(I),(####)
         :Q2(1)=VAL(Q0$,2)+1: REM % array Q2() will hold the numbered line sta
         tements
      540 I2=I2+1
         :IF VAL(Q2$(I2),2)=0  THEN  570
         :I=I+1
      550 IF G1=0  THEN Q2(I2)=VAL(Q2$(I2),2)+1
         :IF G1=1  THEN Q2(I2+1)=VAL(Q2$(I2),2)+1
         :Q=VAL(Q2$(I2),2)+1
         :CONVERT Q  TO Z$(I),(####)
      560 GOTO  540

      570 I3=I3+1
         :IF VAL(Q3$(I3),2)=0  THEN  620
      580 I=I+1
      590 Q=VAL(Q3$(I3),2)
         :CONVERT Q  TO Z$(I),(####)
      600 GOTO  570

      610 REM %

         RE-DIMENSIONING AND SORTING

      620 MAT RE DIM Q2(E5),Z$(I)4,L$(I)2,W$(I)2
      630 MAT SORT Z$()  TO W$(), L$()
      640 H,H1,H3,H4=0
      650 REM %

         LOCATING THE STRING A$ BETWEE

         N EACH Z$() ARRAY
      660 H=H+1
         :CONVERT Z$(VAL(L$(H),2))  TO H3
         :IF H3=VAL(Q1$,2)  THEN  280
         :H1=H
         :IF H1 ]E5  THEN H1=E5
         :CONVERT Z$(VAL(L$(H+1),2))  TO H4
         :I=0
      670 I=I+1
         :IF Q2(I)=H3  THEN A$=STR(X$(),H3+3,(H4-H3)-3)
         :IF Q2(I)=H3  THEN  690
```

PROGRAM LISTING

```
   :IF I[ ]E5  THEN  670
   :A$=STR(X$(),H3+1,(H4-H3)-1)
680 REM %

   LOCATING THE LINE NUMBER THAT

    GOES WITH EACH STATEMENT A$
690 IF H1[ ]E5  THEN  710
700 IF H3]=Q2(H1)  THEN A1$=STR(X$(),Q2(H1)+1,2)
   :IF H3]=Q2(H1)  THEN  760
   :H1=H1-1
   :GOTO  700

710 IF H3]=Q2(H1) AND H3[Q2(H1+1)  THEN A1$=STR(X$(),Q2(H1)+1,2)
   :IF H3]=Q2(H1) AND H3[Q2(H1+1)  THEN  760
   :H1=H1-1
   :IF H1=0  THEN  720
   :GOTO  710

720 SELECT PRINT 005(80)
   :PRINT HEX(03);AT(12,12);"ERROR, (P56) at line No. 640, 'this can be
   caused by a protected program."
730 IF D7$="216"  THEN $CLOSE/216
   :IF D7$="215"  THEN $CLOSE/215
   :IF D7$="204"  THEN $CLOSE/204
   :IF D7$="211"  THEN $CLOSE/211
740 FOR I=1  TO 12000
   :NEXT I
   :RETURN  CLEAR  ALL

   :LOAD T "BASchart"

750 REM %

   GO TO THE STRING CONVERSION T

   HEN THE MAIN BODY
760 UNPACK(####) A1$  TO A1
   :Q,I,I2,I3=0
   :A1$=HEX(00)
   :Q2$(),Q3$()=ALL(00)
770 REM %

   shift the statement to the le

   ft, if needed
   :A1$=A$
   :I2=LEN(A1$)
780 Q=Q+1
   :IF Q=125 THEN  790
   :IF STR(A1$,Q,1)=HEX(20)  THEN  780
   :IF STR(A1$,1,1)[ ]HEX(20)  THEN  800
```

PROGRAM LISTING

```
   :A$=STR(A1$,Q,I2-Q)
   :GOTO 800

790 A$="% COMMENT, THIS LINE CANNOT BE PROCESSED"
800 A1$=HEX(00)
   :I2,Q=0
810 REM %

   identify the first byte (comm

   and) in the statement
820 I=0
830 I=I+1
   :HEXUNPACK STR(A$,1,1) TO Q6$
   :IF Q6$[ ]Q5$(I) THEN 850
   :A1$=A$
   :I2=LEN(A1$)
   :I3=LEN(Q4$(I))
   :IF I2+I3]123 THEN I2=I2-(I3+1)
840 IF I[83 THEN A$=Q4$(I)&" "&STR(A1$,2,I2+I3)
   :ELSE A$=Q4$(I)&STR(A1$,2,I2+I3)
   :Q6$=HEX(00)
   :GOTO 860

850 Q6$=HEX(00)
   :IF I[ ]123 THEN 830
860 A1$=HEX(00)
   :I,I2,I3=0
870 REM %

   identify any referenced line

   number in the statement
880 X1=0
   :Q=POS(A$=HEX(FF))
   :IF Q[122 THEN 890
   :X1=1
   :A1$=A$
   :A$=STR(A1$,1,Q-1)
   :A1$=HEX(00)
   :Q=0
890 IF Q=0 THEN 940
   :A1$=A$
   :I2=LEN(A1$)
   :IF I2-Q]1 THEN 900
   :I=Q
   :GOTO 910

900 I=POS(STR(A$,Q+1,I2-Q)=HEX(20FF))+Q
910 HEXUNPACK STR(A$,Q+1,2) TO Q6$
   :IF I=Q THEN A$=STR(A1$,1,Q-1)&Q6$
920 IF I[ ]Q THEN A$=STR(A1$,1,Q-1)&Q6$&STR(A1$,I,I2-(I-1))
```

PROGRAM LISTING

```
        :Q6$,A1$=HEX(00)
        :I,I2,Q=0
        :GOTO  880

      930 REM %

         identify all HEX coded bytes

         in the statement
      940 Q=0
         :Q6$=HEX(00)
         :I=I+1
         :IF I]123  THEN  980
         :HEXPACK Q6$ FROM Q5$(I)
      950 Q=POS(A$=Q6$)
         :IF Q=0  THEN  940
         :A1$=A$
         :I2=LEN(A1$)-Q
      960 IF I[83  THEN  970
         :IF I2[1  THEN A$=STR(A1$,1,Q-1)&Q4$(I)
         :ELSEA$=STR(A1$,1,Q-1)&Q4$(I)&STR(A1$,Q+1,I2)
         :Q,I2=0
         :A1$=HEX(00)
         :GOTO  940

      970 IF I2[1  THEN A$=STR(A1$,1,Q-1)&Q4$(I)
         :ELSEA$=STR(A1$,1,Q-1)&" "&Q4$(I)&" "&STR(A1$,Q+1,I2)
         :Q,I2=0
         :A1$=HEX(00)
         :GOTO  940

      980 I=0
         :Q=POS(A$=HEX(8A))+POS(A$=HEX(8B))+POS(A$=HEX(D3))+POS(A$=HEX(D2))+
         POS(A$=HEX(F5))+POS(A$=HEX(DC))
         :IF Q[]0  THEN  940
         :CONVERT A1  TO A1$,(####)
      990 IF A1]=N2 AND A1[=N3  THEN D8$=D7$
         :ELSED8$="005"
         :IF A1]N3  THEN 1310
         :Q6$=A$
         :A$=A1$&" "&Q6$
         :A1$,Q6$=HEX(00)
         :IF D8$="005"  THEN 1100
         :SELECT PRINT 005(80)
     1000 IF T=1  THEN 1020
         :PRINT HEX(03)
         :T=1
         :IF A1[]N2  THEN 1020
         :SELECT PRINT [D8$] (130)
         :PRINT HEX(0C)
         :PRINT "////////////////////////////////////////////////////////////////
         ////////////////////////////////////////////////////////////////"
```

PROGRAM LISTING

```
1010 PRINT "                                                    PARTIAL
     FLOW-CHART OF ";T$
    :PRINT "                                                    FROM LI
    NES ";N2;" TO ";N3
1020 SELECT PRINT 005(80)
    :PRINT HEX(06);AT(6,0);STR(A1$,1,80)
1030 PRINT AT(3,0);" [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ NOW READING ]]]]]]]]]]
    ]]]]]]]]]]
```

```
10/25/82      BASCHART        PAGE 10
PROGRAM LISTING

        :GOSUB '66
        :IF U=1  THEN GOSUB 3750
     1140 GOSUB '15
        :IF E9=1  THEN 1140
        :IF U6=1  THEN GOSUB '66
     1150 IF A2$="STOP" OR A2$="END"  THEN L3=1
        :RETURN

     1160 U=0
        :IF C$="N"  THEN M=1
        :IF M=1  THEN 1230
        :IF XO=1  THEN GOSUB '81
        :IF YO=1  THEN GOSUB '83
        :GOSUB '66
        :PRINT
        :PRINT
        :PRINT
     1170 PRINT "          $$$$
```

PROGRAM LISTING

```
    :PRINT
1250 PRINT  STR(A$,J)
    :PRINT
    :PRINT "********************"
    :PRINT "*    SUBROUTINE    *"
    :PRINT "*    ";STR(A$,1,J-1);"         *"
    :PRINT "********************"
    :PRINT "        !"
1260 E=0
    :RETURN

1270 GOSUB '66
    :IF U=1  THEN GOSUB 3750
1280 GOSUB '15
    :IF A2$="STOP" OR A2$="END"  THEN L3=1
    :IF E9=1  THEN 1280
    :IF U6=1  THEN GOSUB '66
    :IF A2$="END"  THEN GOTO 1300
1290 IF A2$[ ]"END"  THEN L2=1
    :IF L2=1  THEN RETURN
1300 PRINT
    :PRINT "COMMENT: THERE IS AN /END/ COMMAND IN THIS SUBROUTINE."
    :PRINT
    :W3=W3+1
    :RETURN

1310 IF X0=1  THEN GOSUB '81
    :IF Y0=1  THEN GOSUB '83
    :GOSUB '66
    :PRINT "//////////////////////////////////////////////////////////////////
    //////////////////////////////////////////////////////////////////"
    :PRINT
1320 IF N3[ ]9999  THEN 1390
1330 IF J1[ ]1  THEN 1340
    :PRINT
    :PRINT
    :PRINT "NOTICE:   MULTI-STATEMENT LINE NUMBERS WERE USED IN THIS PROG
    RAM"
1340 IF L3=1  THEN 1350
    :PRINT
    :PRINT
    :PRINT "COMMENT,   THERE WAS NO /END/ OR /STOP/ STATEMENT IN THIS PRO
    GRAM"
    :W3=W3+1
1350 IF W3=0  THEN 1370
    :PRINT
    :PRINT
    :IF W3[ ]1  THEN 1360
    :PRINT "NOTICE:   THERE WAS 1 /COMMENT/ GIVEN DURING THIS RUN"
    :GOTO 1370

1360 PRINT "NOTICE:   THERE WERE";W3;" /COMMENTS/ GIVEN DURING THIS RUN"
```

PROGRAM LISTING

```
1370 PRINT
1380 IF G=0  THEN 1390
    :PRINT
    :PRINT "COMMENT:   THIS PROGRAM CONTAINS 'GOTO' STATEMENTS"
    :IF G=1  THEN PRINT "           THERE WAS 1 /GOTO/ DETECTED DURING TH
    IS RUN"
    :IF G]1  THEN PRINT "           THERE WERE";G;"/GOTO'S/ DETECTED DURI
    NG THIS RUN"
    :PRINT
    :PRINT
1390 IF D7$="215"  THEN $CLOSE/215
    :IF D7$="216"  THEN $CLOSE/216
    :IF D7$="204"  THEN $CLOSE/204
    :IF D7$="211"  THEN $CLOSE/211
    :SELECT PRINT 005(80)
1400 I,J=0
    :PRINT HEX(07)
    :FOR I=10  TO 1200
    :NEXT I
    :FOR I=1  TO 3
    :PRINT HEX(07)
    :FOR J=1  TO 800
    :NEXT J,I
    :FOR I=1  TO 400
    :NEXT I
    :PRINT HEX(07)
    :FOR I=1  TO 2400
    :NEXT I
    :FOR I=1  TO 2
    :PRINT HEX(07)
    :FOR J=1  TO 1000
    :NEXT J,I
1410 PRINT HEX(03);AT(20,7);BOX(1,56)
    :Z1$=HEX(0D)
    :PRINT AT(5,8);"Processing + Printing completed.."
    :PRINT AT(20,8);"Was it your intention to run this program again? (Y/
    N)"
    :KEYIN Z1$
    :IF Z1$="Y"  THEN 1420
    :IF Z1$="N"  THEN 1430
    :GOTO 1410

1420 RETURN  CLEAR  ALL

    :Z1=1
    :LOAD T "BASchart"

1430 $PSTAT=" "
    :RETURN  CLEAR  ALL

    :COM CLEAR
```

10/25/82 BASCHART PAGE 13
PROGRAM LISTING

```
        :LOAD T "START"

        :ERRORPRINT HEX(03);AT(15,8);"UNABLE TO LOAD THE START PROGRAM"
        :STOP

        :END

      1440 REM %

        SUBROUTINE (DEFFN'15) .......

        .....................................
      1450 REM search statement string for / locate and return position of
      1460 REM ',' ' ' 'THEN' /compare statement against
      1470 REM (FOR,NEXT,ELSE,REM)
      1480 DEFFN'15
        :T2$=A$
        :E9=0
        :F6=1
        :J=0
        :GOSUB '96
        :IF P=1  THEN RETURN
        :E2=POS(STR(A$,J)=HEX(20))
        :N$=STR(A$,J,E2-J+1)
      1490 IF T2$=" "  THEN RETURN
      1500 J=0
        :GOSUB '96
        :J=POS(STR(T2$,J)=HEX(20))
        :GOSUB '96
        :B1=J
        :J=POS(STR(T2$,J)=HEX(20))
        :IF J]123 OR J[2  THEN J=2
        :F$=STR(T2$,B1,J-1)
      1510 IF STR(F$,1,3)="REM" OR STR(F$,1,1)="%" AND Z1$="N"  THEN RETURN
        :IF F6=0  THEN A$=T2$&S9$
        :T2$=" "
        :GOTO 1520

      1520 IF E[=4  THEN 1620
        :IF F4[]0  THEN 1530
        :W1=W1+1
        :GOSUB 3820
        :F4=1
      1530 J=0
        :GOSUB '96
        :IF P=1  THEN RETURN
        :B=J
        :J=POS(STR(A$,J)=HEX(20))
        :E1=J
        :GOSUB '96
        :B1=J
        :J=POS(STR(A$,J)=HEX(20))
```

PROGRAM LISTING

```
    :IF J]123 OR J[2  THEN J=2
1540 E3=J
    :K2$=STR(A$,B1,E3-1)
1550 IF K2$[ ]"IF"  THEN 1580
    :F5=1
1560 MAT SEARCH STR(A$,J),=HEX(205448454E)  TO A1$
    :J3=VAL(A1$,2)
    :A1$=ALL(00)
    :IF J3[=0  THEN GOSUB '19
1570 IF F5=0  THEN RETURN
    :F5=0
    :J=J3+5
    :T1$=" "
    :GOSUB '95
1580 IF K2$[ ]"FOR" AND K2$[ ]"NEXT"  THEN 1590
    :T1$=S9$
    :GOSUB '95
1590 IF K2$="FOR"  THEN E=E+1
    :IF K2$="NEXT"  THEN E=E-1
    :J0=0
    :J0=POS(B2$=",")
    :IF J0=0 OR K2$[ ]"NEXT"  THEN 1600
    :A1$=B2$
    :B2$=STR(A1$,1,J0-1)
    :A$=N$&" "&K2$&" "&STR(A1$,J0+1)
    :E9=1
1600 A2$=K2$
    :IF E[=4  THEN 1610
    :IF E9=1  THEN PRINT N$;" ";A2$;" ";B2$
    :ELSEPRINT STR(A$,1,124)
    :GOTO 1630

1610 E=E+1
    :F4=0
    :A2$=K2$
    :IF K2$="FOR" OR K2$="NEXT"  THEN J=E3
    :GOSUB 1680
    :GOTO 1630

1620 GOSUB '10
1630 IF A2$=" FOR"  THEN A2$="FOR"
    :IF A2$=" NEXT"  THEN A2$="NEXT"
    :IF A2$="FOR"  THEN F2$=STR(B2$,1,POS(STR(B2$,1)="=")-1)
    :GOSUB '98
    :F6=0
1640 GOTO 1490

1650 REM %
```

PROGRAM LISTING

```
        SUBROUTINE (DEFFN '10) ......

        ....................................
     1660 REM locate / compare (FOR,NEXT,REM,GOTO,GOSUB)
     1670 DEFFN'10
         :J=0
         :GOSUB '96
         :B=J
         :J=POS(STR(A$,J)=HEX(20))
         :E1=J
         :GOSUB '96
         :IF P=1  THEN RETURN
         :B1=J
         :J=POS(STR(A$,J)=HEX(20))
         :IF J]123 OR J[2  THEN J=2
         :E3=J
         :A2$=STR(A$,B1,E3-1)
     1680 S=0
         :IF A2$[ ]"FOR"  THEN 1690
         :T1$=S9$
         :GOSUB '95
         :GOSUB '30
         :RETURN

     1690 IF A2$[ ]"NEXT"  THEN 1710
         :IF E9=1  THEN 1700
         :T1$=S9$
         :GOSUB '95
         :J0=0
         :J0=POS(B2$=",")
         :IF J0=0  THEN 1700
         :A1$=B2$
         :B2$=STR(A1$,1,J0-1)
         :A$=N$&" "&A2$&" "&STR(A1$,J0+1)
         :E9=1
         :A1$=HEX(00)
     1700 GOSUB '35
         :RETURN

     1710 IF STR(A2$,1,1)[ ]"%" AND STR(A2$,1,3)[ ]"REM"  THEN 1720
         :B2=B1
         :B2$=STR(A$,B2)
         :GOSUB '75
         :RETURN

     1720 IF A2$[ ]"GOTO"  THEN 1730
         :T1$=" "
         :GOSUB '95
         :D2$="GOTO"
         :GOSUB '45
         :RETURN
```

10/25/82 BASCHART PAGE 16
PROGRAM LISTING

```
1730 IF STR(A2$,1,5)[]"GOSUB"  THEN 1740
    :T1$=" "
    :GOSUB '95
    :D2$="GOSUB"
    :GOSUB '45
    :RETURN

1740 IF A2$[]"ON"  THEN 1800
    :T1$=" "
    :J=B1-1
    :GOSUB '95
    :MAT SEARCH A$,="GOTO"  TO A1$
    :IF VAL(A1$,2)=0  THEN MAT SEARCH A$,="GOSUB"  TO A1$
    :J=VAL(A1$,2)
    :A1$=ALL(00)
    :IF J[]0  THEN 1750
    :J=E3
    :GOTO 1800

1750 C2$=STR(A$,E0+B2,J-(B2+E0+1))
    :J=POS(-STR(A$,1,POS(-STR(A$,1,J-1)[]HEX(20)))=HEX(20))
    :GOSUB '95
    :J0=0
1760 J0=J0+1
    :IF STR(B2$,J0,1)=" "  THEN 1760
1770 IF STR(B2$,J0,4)[]"GOTO"  THEN 1780
    :T1$=S9$
    :J=B2+E0-1
    :GOSUB '95
    :D2$="GOTO"
    :GOTO 1790

1780 IF STR(B2$,J0,5)[]"GOSUB"  THEN 1800
    :T1$=S9$
    :J=B2+E0-1
    :GOSUB '95
    :D2$="GOSUB"
1790 GOSUB '40
    :RETURN

1800 IF S=0  THEN GOSUB '20
    :RETURN

1810 REM %

    SUBROUTINE (DEFFN'20) .......

    .....................................
1820 REM compare (IF,PRINT,INPUT,GOSUB,GOTO,STOP,END,RETURN)
1830 DEFFN '20
1840 IF A2$[]"IF"  THEN 1950
    :J=B1+E3-1
```

PROGRAM LISTING

```
        :GOSUB '96
        :IF P=1  THEN RETURN
        :B2=J
        :GOSUB 1560
        :IF J3=0  THEN RETURN
        :J=J3+B2
        :C2$=STR(A$,B2,J3-1)
        :T1$=" "
        :GOSUB '95
     1850 D2$=B2$
        :J=0
     1860 J=J+1
        :IF STR(D2$,J,1)=" "  THEN 1860
     1870 B2$=STR(D2$,J,POS(STR(D2$,J)=HEX(20))-1)
        :D2$=" "
     1880 IF B2$="PRINT" OR B2$="INPUT" OR B2$="PRINTUSING" OR B2$="KEYIN"
        OR B2$="LINPUT" OR B2$="GOTO" OR B2$="GOSUB"  THEN D2$=B2$
     1890 IF B2$[ ]"PRINT" AND B2$[ ]"INPUT" AND B2$[ ]"PRINTUSING" AND B2$[ ]"KEY
        IN" AND B2$[ ]"LINPUT" AND B2$[ ]"GOTO" AND B2$[ ]"GOSUB"  THEN 1900
        :J=B2+ED-1
        :GOTO 1930

     1900 IF B2$]=":"  THEN 1920
        :CONVERT B2$  TO N
        :ERRORN=0
        :GOTO 1920

     1910 N=0
        :D2$="GOTO"
        :J=B2-1
        :GOTO 1930

     1920 N=0
        :D2$=" "
        :J=B2-1
     1930 T1$=S9$
        :GOSUB '95
        :GOSUB '60
     1940 RETURN

     1950 IF A2$="ELSE" OR A2$="ERROR"  THEN GOSUB '65
        :IF A2$="LOAD" OR A2$="STOP" OR A2$="END"  THEN GOSUB '50
        :IF A2$="RETURN" AND STR(B2$,1,5)[ ]"CLEAR"  THEN GOSUB '85
        :IF S[ ]0  THEN RETURN
     1960 T1$=S9$
        :IF STR(A2$,1,5)[ ]"PRINT" AND STR(A2$,1,10)[ ]"PRINTUSING"  THEN 1980
        :J4=J+E1-1
     1970 J4=J4+1
        :IF STR(A$,J4,1)=" " AND J4[81  THEN 1970
        :IF J4=81  THEN B2$="1 BLANK LINE"
        :IF J4[ ]81  THEN GOSUB '95
        :GOSUB '82
```

```
    :RETURN

1980 IF STR(A2$,1,5)[]"INPUT" AND STR(A2$,1,5)[]"KEYIN" AND STR(A2$,1,6)
    []"LINPUT"  THEN J=B1-1
    :GOSUB '95
1990 GOSUB '80
    :RETURN

2000 REM %

    SUBROUTINE (DEFFN'98) .......

    .....................................
2010 REM compare (FOR,ELSE,NEXT)
2020 DEFFN'98
2030 IF A2$[]"FOR"  THEN 2040
    :C=C+1
    :IF C[51  THEN L1$(C)="FOR "&F2$
    :IF C[51  THEN L2$(C)=STR(A$,B,E1-B)
    :GOTO 2060

2040 IF A2$[]"NEXT"  THEN 2060
    :IF C[51  THEN A2$="NEXT "&B2$
    :L3$="FOR "&B2$
    :IF C[1  THEN GOSUB '99
    :IF C[1  THEN 2060
    :IF C[51 AND L1$(C) [] L3$  THEN GOSUB '99
2050 C=C-1
2060 FOR J5=B  TO E1-B
    :IF STR(A$,J5,1)["0" OR STR(A$,J5,1)]"9"  THEN GOSUB '92
    :NEXT J5
2070 CONVERT STR(A$,B,E1-B)  TO T1
    :IF T1[=L AND F6=1  THEN GOSUB '91
    :L=T1
    :RETURN

2080 REM %

    SUBROUTINE (DEFFN'60) .......

    .....................................
2090 REM print diamond symbol / compare (GOTO) / print-
2100 REM YES or NO option / compare (PRINT,INPUT,GOSUB) / print-
2110 REM PRINT or INPUT
2120 DEFFN'60
    :N1,N=0
    :IF X1=1  THEN GOSUB 2390
2130 CONVERT B2$  TO N1
    :ERRORN,N1=0
    :GOTO 2150

2140 CONVERT N$  TO N
```

PROGRAM LISTING

```
        :ERRORN1,N=0
        :GOTO 2150

     2150 K=E*20
        :IF XO=1  THEN GOSUB '81
        :IF YO=1  THEN GOSUB '83
        :S=1
        :FOR S3=1  TO 2
        :GOSUB '90
        :PRINT TAB(K);"        !"
        :NEXT S3
        :IF E=0  THEN K=1
     2160 GOSUB '90
        :PRINT TAB(K);"       /*";N$
        :GOSUB '90
        :PRINT TAB(K);"      /  *"
        :GOSUB '90
        :PRINT TAB(K);"     /    *";
        :IF D2$[ ]"GOTO"  THEN PRINT " "
        :IF D2$[ ]"GOTO"  THEN 2170
        :IF N1[N  THEN PRINT TAB(K+36);"  /!"
        :ELSEPRINT " "
     2170 GOSUB '90
        :PRINT TAB(K);"    /  IF  *";C2$;
        :IF D2$[ ]"GOTO"  THEN PRINT " "
        :IF D2$[ ]"GOTO"  THEN 2180
        :IF N1[N  THEN PRINT TAB(K+36);"   !"
        :ELSEPRINT " "
     2180 GOSUB '90
        :PRINT TAB(K);"   /  THEN  ";
     2190 IF D2$[ ]"GOTO"  THEN 2240
        :G=G+1
        :PRINT "*";
        :IF N1[N  THEN PRINT TAB(K+36);"   !"
        :ELSEPRINT " "
        :GOSUB '93
        :PRINT TAB(K);"   *  ";D2$;"  /       [YES]";
        :IF N1[N  THEN PRINT TAB(K+36);"   !"
        :ELSEPRINT " "
        :GOSUB '90
     2200 PRINT TAB(K);"    * ";B2$;TAB(K+11);
        :PRINT "/ ------------------]-------"
        :GOSUB '90
        :PRINT TAB(K);"     *    /";
        :IF N1]N  THEN PRINT TAB(K+38);" !"
        :ELSEPRINT " "
        :GOSUB '90
     2210 PRINT TAB(K);"      *  /";
        :IF N1]N  THEN PRINT TAB(K+38);" !"
        :ELSEPRINT " "
        :GOSUB '90
     2220 PRINT TAB(K);"       */";
```

```
        :IF N1]N  THEN PRINT TAB(K+38);" !"
        :ELSEPRINT " "
        :IF E=0  THEN K=0
        :GOSUB '90
        :PRINT TAB(K);"        !";
        :IF N1]N  THEN PRINT TAB(K+38);"  V"
        :ELSEPRINT " "
        :GOSUB '90
     2230 PRINT TAB(K);"       [NO]"
        :GOTO 2380

     2240 IF D2$[ ]"KEYIN" AND D2$[ ]"LINPUT" AND D2$[ ]"INPUT"  THEN 2250
        :PRINT "*"
        :GOSUB '93
        :PRINT TAB(K);"   *  ";D2$;" / ";B2$
        :GOSUB '90
        :PRINT TAB(K);"     *      /"
        :GOTO 2370

     2250 IF D2$[ ]"PRINT" AND D2$[ ]"PRINTUSING"  THEN 2290
     2260 PRINT "*"
        :GOSUB '93
        :PRINT TAB(K);"   *        / [YES]    /===============/"
        :GOSUB '90
        :PRINT TAB(K);"    *      /=========] /               /"
     2270 GOSUB '90
        :IF D2$="PRINT"  THEN PRINT TAB(K);"     *    /          (     PRINT
            ( ";B2$
        :ELSEPRINT TAB(K);"     *    /          (   PRINTUSING  (  ";B2$
        :GOSUB '90
        :PRINT TAB(K);"     * /            *              *"
        :GOSUB '90
        :PRINT TAB(K);"       */              ================="
     2280 GOSUB '90
        :PRINT TAB(K);"       !"
        :GOSUB '90
        :PRINT TAB(K);"      [NO]"
        :GOTO 2380

     2290 IF STR(D2$,1,5)[ ]"GOSUB"  THEN 2330
     2300 PRINT "*"
        :GOSUB '93
        :PRINT TAB(K);"   *        / [YES] ..............."
        :GOSUB '90
        :PRINT TAB(K);"    *      /=========]:             :"
     2310 GOSUB '90
        :PRINT TAB(K);"     *    /          : ";D2$;" ";B2$;TAB(K+36);":"
        :GOSUB '90
        :PRINT TAB(K);"      * /           :             :"
        :GOSUB '90
        :PRINT TAB(K);"       */            :.............:"
     2320 GOSUB '90
```

PROGRAM LISTING

```
        :PRINT TAB(K);"       !"
        :GOSUB '90
        :PRINT TAB(K);"      [NO]"
        :GOTO 2380

    2330 IF STR(B2$,1,6)[]"RETURN" AND STR(B2$,1,7)[]" RETURN"  THEN 2360
        :PRINT "*"
        :GOSUB '93
        :PRINT TAB(K);"   *        / [YES]  ::::::::::::::::"
        :GOSUB '90
        :PRINT TAB(K);"    *      /=========]::           ::"
    2340 GOSUB '90
        :PRINT TAB(K);"     *    /           :: RETURN   ::"
        :GOSUB '90
        :PRINT TAB(K);"      * /            ::          ::"
        :GOSUB '90
        :PRINT TAB(K);"       */            ::::::::::::::::"
    2350 GOSUB '90
        :PRINT TAB(K);"       !"
        :GOSUB '90
        :PRINT TAB(K);"      [NO]"
        :GOTO 2380

    2360 PRINT B2$
        :GOSUB '93
        :PRINT TAB(K);"   *        /"
        :GOSUB '90
        :PRINT TAB(K);"    *      /"
    2370 GOSUB '90
        :PRINT TAB(K);"     *    /"
        :GOSUB '90
        :PRINT TAB(K);"      * /"
        :GOSUB '90
        :PRINT TAB(K);"       */"
    2380 IF E=0  THEN K=0
        :GOSUB '8
        :RETURN

    2390 PRINT
        :PRINT "COMMENT,   THIS /IF...THEN/ STATEMENT MAY BE TOO LONG TO PROC
        ESS TO ITS EXTENT"
        :PRINT
        :W3=W3+1
        :RETURN

    2400 REM %

        SUBROUTINE (DEFFN'30) .......

        .................................
    2410 REM print diamond (decision) / compare (FOR)
    2420 DEFFN'30
```

PROGRAM LISTING

```
2430 K=E*20
    :IF XO=1  THEN GOSUB '81
    :IF YO=1  THEN GOSUB '83
    :S=1
2440 GOSUB '90
    :PRINT TAB(K);"        !"
    :GOSUB '90
    :PRINT TAB(K);"        !"
    :GOSUB '90
    :PRINT TAB(K);"       /*";N$
    :GOSUB '90
    :PRINT TAB(K);"      /  *"
2450 GOSUB '90
    :PRINT TAB(K);"     /    *"
    :GOSUB '90
    :PRINT TAB(K);"    /      *"
    :IF A2$="FOR"  THEN A2$=" FOR"
    :GOSUB '90
2460 PRINT TAB(K);"   / ";A2$;TAB(K+12);"* ";B2$
    :GOSUB '93
    :PRINT TAB(K);"   *        /-------]--------"
2470 GOSUB '90
    :PRINT TAB(K);"    *      /";TAB(K+28);"!"
    :GOSUB '90
    :PRINT TAB(K);"     *    /";TAB(K+27);" V"
2480 GOSUB '90
    :PRINT TAB(K);"      *  /";TAB(K+28);"!"
    :GOSUB '90
    :PRINT TAB(K);"       */";TAB(K+28);"!"
2490 E=E+1
    :FO(E)=1
    :RETURN

2500 REM %

    SUBROUTINE (DEFFN'35) .......

    .................................
2510 REM print diamond (decision) / compare (NEXT)
2520 DEFFN'35
2530 IF XO=1  THEN GOSUB '81
    :IF YO=1  THEN GOSUB '83
    :IF E=0  THEN RETURN
    :FO(E)=0
    :E=E-1
    :K=E*20
    :S=1
2540 GOSUB '90
    :PRINT TAB(K);"       /*";TAB(K+28);"!"
    :GOSUB '90
    :PRINT TAB(K);"      /  *";TAB(K+28);"!"
    :GOSUB '90
```

```
2550 PRINT TAB(K);"     /    *";TAB(K+28);"!"
    :GOSUB '90
    :PRINT TAB(K);"    /      *";TAB(K+28);"!"
2560 GOSUB '93
    :PRINT TAB(K);"   /        *-------[--------"
    :GOSUB '90
    :IF A2$="NEXT"  THEN A2$=" NEXT"
2570 PRINT TAB(K);"   * ";A2$;TAB(K+12);"/ ";B2$
    :GOSUB '90
    :PRINT TAB(K);"    *      /"
    :GOSUB '90
2580 PRINT TAB(K);"     *    /"
    :GOSUB '90
    :PRINT TAB(K);"      *  /"
    :GOSUB '90
    :PRINT TAB(K);"       */"
    :GOSUB '8
    :RETURN

2590 REM %

    SUBROUTINE (DEFFN'40) .......

    .....................................
2600 REM locate position in string statement of ',' / print -lines
2610 REM and arrows / print diamond with ON / compare (GOTO)
2620 DEFFN'40
    :IF X1=1  THEN GOSUB 2820
2630 W,J,J0=0
2640 V2=J0
    :W=W+1
    :J=0
2650 J=POS(STR(B2$,V2+1)=",")
    :J0=J0+J
    :IF J0-V2-1=0 AND J]0  THEN V(W)=0
    :IF J0-V2-1=0 AND J]0  THEN 2640
    :IF J]0  THEN CONVERT STR(B2$,V2+1,J0-V2-1)  TO V(W)
2660 IF J[]0  THEN 2640
    :CONVERT STR(B2$,V2+1,LEN(B2$)-V2)  TO V(W)
2670 S=1
2680 IF X0=1  THEN GOSUB '81
    :IF Y0=1  THEN GOSUB '83
    :K=E*20
    :W2=W-1
    :GOSUB '90
    :PRINT TAB(K);"        -]";
    :FOR Z=1  TO W2
    :PRINT "----";
    :NEXT Z
2690 PRINT " "
    :K3=W2*4+K
    :GOSUB '90
```

```
    :PRINT TAB(K3);"          !"
    :GOSUB '90
    :PRINT TAB(K3);"          /*";N$
    :GOSUB '90
    :PRINT TAB(K3);"         /  *"
2700 GOSUB '90
    :PRINT TAB(K3);"        /    *"
    :GOSUB '90
    :PRINT TAB(K3);"       /      *"
    :GOSUB '90
    :PRINT TAB(K3);"      /  ON    *";C2$
2710 GOSUB '90
    :PRINT TAB(K3);"      *  ";D2$;TAB(K3+15);"/"
    :GOSUB '90
    :PRINT TAB(K3);"       *      /"
    :GOSUB '90
2720 PRINT TAB(K3);"        *    /"
    :GOSUB '90
    :PRINT TAB(K3);"         *  /"
    :GOSUB '90
    :PRINT TAB(K3);"          */"
    :GOSUB '90
    :PRINT TAB(K3);"          !"
2730 GOSUB '93
    :PRINT TAB(K);"          -";
    :FOR Z=1  TO W2
    :PRINT "--------";
    :NEXT Z
    :PRINT " "
    :GOSUB '90
    :PRINT TAB(K);"    ";
    :FOR Z=1  TO W
    :PRINT "     ";Z;
    :NEXT Z
2740 GOSUB '90
    :PRINT TAB(K);"   ";
    :FOR Z=1  TO W
    :PRINT "        ";
    :NEXT Z
    :PRINT " "
2750 GOSUB '90
    :PRINT TAB(K);"     ";
    :FOR Z=1  TO W
    :PRINT "   -----";
    :NEXT Z
    :PRINT " "
    :GOSUB '90
    :FOR Z=1  TO W
    :E$="       "
2760 CONVERT V(Z)  TO E$,(#####)
    :PRINT TAB(Z*8+K);STR(E$,1,5);
    :NEXT Z
```

PROGRAM LISTING

```
2770 PRINT " "
    :GOSUB '90
    :PRINT TAB(K);"     ";
    :FOR Z=1  TO W
    :PRINT "   -----";
    :NEXT Z
    :PRINT " "
    :FOR F=1  TO 2
    :GOSUB '90
    :PRINT TAB(K);"    ";
    :FOR Z=1  TO W
    :PRINT "        ";
    :NEXT Z
2780 PRINT " "
    :NEXT F
    :GOSUB '90
    :PRINT TAB(K);"          -";
    :FOR Z=1  TO W2
    :PRINT "--------";
    :NEXT Z
    :PRINT " "
    :GOSUB '90
    :PRINT TAB(K3);"           "
    :GOSUB '90
    :PRINT TAB(K);"        --[";
2790 FOR Z=1  TO W2
    :PRINT "----";
    :NEXT Z
    :PRINT "!"
2800 IF D2$[ ]"GOTO"  THEN 2810
    :G=G+1
    :B2$=N$
    :GOSUB '4
    :RETURN

2810 GOSUB '8
    :RETURN

2820 PRINT
    :PRINT "COMMENT,   THIS /ON..GOSUB(GOTO)/ STATEMENT MAYBE TOO LONG TO
     PROCESS TO ITS EXTENT"
    :PRINT
    :A1$=B2$
    :J=POS(-B2$=",")
    :B2$=STR(A1$,1,J-1)
    :A1$=HEX(00)
    :W3=W3+1
    :RETURN

2830 REM %
```

10/25/82 BASCHART PAGE 26
PROGRAM LISTING

```
    SUBROUTINE (DEFFN'45) .......

    ................................
2840 REM print - block around GOTO / compare (GOTO)
2850 DEFFN'45
2860 S=1
    :IF XO=1  THEN GOSUB '81
    :IF YO=1  THEN GOSUB '83
    :K=E*20
    :GOSUB '90
    :PRINT TAB(K);"        !"
    :GOSUB '90
    :PRINT TAB(K);"        !"
    :IF E=0  THEN K=1
2870 GOSUB '90
    :PRINT TAB(K);".............";N$
    :GOSUB '90
    :PRINT TAB(K);":             :"
    :GOSUB '93
    :IF D2$[ ]"GOTO"  THEN 2880
    :D2$=" GOTO"
    :G=G+1
2880 PRINT TAB(K);": ";D2$;" ";B2$;TAB(K+14);":"
    :GOSUB '90
    :PRINT TAB(K);":             :"
    :GOSUB '90
    :PRINT TAB(K);":............:"
    :IF E=0  THEN K=0
2890 IF D2$=" GOTO"  THEN GOSUB '4
    :ELSEGOSUB '8
    :RETURN

2900 REM %

    SUBROUTINE (DEFFN'50) .......

    ................................
2910 REM print - block around STOP or END or LOAD
2920 DEFFN'50
2930 U=1
    :IF A2$[ ]"LOAD"  THEN B2$=" "
    :IF A2$[ ]"LOAD"  THEN 2940
    :J=E3
    :GOSUB '95
2940 S=1
    :IF XO=1  THEN GOSUB '81
    :IF YO=1  THEN GOSUB '83
    :K=E*20
    :GOSUB '90
    :PRINT TAB(K);"        !"
    :GOSUB '90
    :PRINT TAB(K);"        !"
```

PROGRAM LISTING

```
    :GOSUB '90
    :IF E=0  THEN K=1
    :PRINT TAB(K);"            ";N$
    :GOSUB '90
2950 PRINT TAB(K);"  (--------)"
    :GOSUB '90
    :PRINT TAB(K);" (    ";A2$;TAB(K+12);")";B2$
    :GOSUB '90
    :PRINT TAB(K);"  (--------)"
    :GOSUB '90
    :PRINT
    :GOSUB '90
    :PRINT
    :R1=1
    :RETURN

2960 REM %

    SUBROUTINE (DEFFN'65) .......

    ................................
2970 REM print - block around ELSE OR ERROR/ print lines and arrows
2980 DEFFN'65
2990 IF X0=1  THEN GOSUB '81
    :IF Y0=1  THEN GOSUB '83
    :K=E*20
    :GOSUB '90
    :PRINT TAB(K);"    ********** ";N$
    :GOSUB '90
    :IF A2$="ELSE"  THEN PRINT TAB(K);"   *   ELSE    * =======]----"
    :ELSEPRINT TAB(K);"   *  ERROR    * =======]----"
    :GOSUB '90
3000 PRINT TAB(K);"    **********             !"
    :E=E+1
    :IF A2$="ELSE"  THEN U6=1
    :ELSEU6=2
3010 J=E3
    :GOSUB '95
    :A$=N$&" "&B2$
    :S=1
    :E9=1
    :RETURN

3020 REM %

    SUBROUTINE (DEFFN'75) .......

    ................................
3030 REM print the REMARKS statement
3040 DEFFN'75
3050 S=1
    :IF X0=1  THEN GOSUB '81
```

PROGRAM LISTING

```
        :IF YO=1  THEN GOSUB '83
        :K=E*20
        :GOSUB '90
        :PRINT TAB(K);B2$
        :RETURN

    3060 REM %

        SUBROUTINE (DEFFN'81) .......

        .....................................
    3070 REM print line then reference print routine
    3080 DEFFN'81
    3090 K=E*20
        :XO=0
        :IF E=0  THEN K=1
        :GOSUB '90
        :PRINT TAB(K);":.............:"
        :IF E=0  THEN K=0
        :GOSUB '8
        :RETURN

    3100 REM %

        SUBROUTINE (DEFFN'85) .......

        .....................................
    3110 REM print block around RETURN/warning of no stop instr.before subro
        utine
    3120 DEFFN'85
    3130 U=1
        :K=E*20
        :IF XO=1  THEN GOSUB '81
        :IF YO=1  THEN GOSUB '83
        :FOR S3=1  TO 2
        :PRINT TAB(K);"        !"
        :NEXT S3
        :IF E=0  THEN K=1
        :PRINT TAB(K);":::::::::::::::: ";N$
        :PRINT TAB(K);"::           ::"
    3140 PRINT TAB(K);"::   RETURN  ::"
        :PRINT TAB(K);"::           ::"
        :PRINT TAB(K);":::::::::::::::"
        :PRINT
        :S=1
    3150 IF R1[ ]0  THEN RETURN
        :PRINT
        :PRINT "COMMENT,     THERE WAS NO /STOP/ INSTRUCTION BEFORE THIS SUBR
        OUTINE"
        :W3=W3+1
        :PRINT
        :R1=1
```

```
    :RETURN

3160 REM %

    SUBROUTINE (DEFFN'90) .......

    .................................
3170 REM print down line
3180 DEFFN'90
3190 IF E=0  THEN RETURN
3200 FOR K2=1  TO E
    :PRINT TAB(K2*20-12);"!";
    :NEXT K2
    :RETURN

3210 REM %

    SUBROUTINE (DEFFN'91) .......

    .................................
3220 REM print warning of instruction No.
3230 DEFFN'91
3240 J1=1
    :IF T1=L  THEN RETURN
    :PRINT
    :PRINT "COMMENT,  EITHER INSTRUCTION NUMBER ";T1;"OR";L;"  IS OUT OF
    ORDER"
3250 W3=W3+1
    :PRINT
    :RETURN

3260 REM %

    SUBROUTINE (DEFFN'92) .......

    .................................
3270 REM print line sequence error, abort
3280 DEFFN'92
3290 PRINT
    :PRINT "COMMENT,    INSTRUCTION  ";STR(A$,1)
    :PRINT " DOES NOT HAVE A CORRECT INSTRUCTION NUMBER."
    :PRINT
    :W3=W3+1
    :RETURN

3300 REM %

    SUBROUTINE (DEFFN'93) .......

    .................................
3310 REM print up or down arrows
3320 DEFFN'93
```

PROGRAM LISTING

```
3330 IF E=0  THEN RETURN
3340 FOR K2=1  TO E
    :IF FO(K2)=1  THEN PRINT TAB(K2*20-13);"/! ";
    :ELSEPRINT TAB(K2*20-13);" !/";
    :NEXT K2
    :RETURN

3350 REM %

    SUBROUTINE (DEFFN'95) .......

    .................................
3360 REM locate and save line following the word after the value J: REM
    if can't then save the word after J
3370 DEFFN'95
3380 GOSUB '96
    :IF P=1  THEN RETURN
    :B2=J
    :E4=POS(-A$=HEX(0D))
    :IF E4=0  THEN E4=POS(-A$[ ]HEX(20))+1
    :EO=POS(STR(A$,B2)=HEX(20))
    :IF E4]=115  THEN X1=1
    :IF EO+B2]=E4  THEN 3400
    :B2$=STR(A$,EO+B2,E4-(EO+B2))
3390 RETURN

3400 IF B2]=E4  THEN 3410
    :B2$=STR(A$,B2)
    :RETURN

3410 B2$=" "
    :RETURN

3420 REM %

    SUBROUTINE (DEFFN'96) .......

    .................................
3430 REM print warning of statement error, abort
3440 DEFFN'96
3450 J=J+1
    :IF J]124  THEN 3460
    :IF STR(A$,J,1)=" "  THEN 3450
    :IF J[=124  THEN RETURN
3460 PRINT
3470 PRINT "COMMENT,   NOT A COMPLETE BASIC INSTRUCTION"
    :P=1
    :W3=W3+1
    :PRINT A$
    :RETURN

3480 REM %
```

```
    SUBROUTINE (DEFFN'99) .......

    .....................................
3490 REM print line sequence reference error, abort
3500 DEFFN'99
3510 W3=W3+1
    :PRINT
    :PRINT "COMMENT,    THIS PROGRAM HAS A LOGIC ERROR"
    :PRINT "          /";A2$;"/  OF LINE NUMBER  ";N$
    :IF C=0  THEN RETURN
    :PRINT "            DOES NOT MATCH UP WITH /";L1$(C);"/  OF LINE NUMB
    ER ";L2$(C)
3520 RETURN

3530 REM %

    SUBROUTINE (DEFFN'80) .......

    .....................................
3540 REM print block around PROCESS / compare and print (PRINT,INPUT)
3550 DEFFN'80
    :K=E*20
    :IF Y0=1  THEN GOSUB '83
3560 IF X0[]0  THEN 3580
    :FOR S3=1  TO 2
    :GOSUB '90
    :PRINT TAB(K);"        !              "
    :NEXT S3
    :IF E=0  THEN K=1
    :GOSUB '90
3570 PRINT TAB(K);".............";N$
    :GOSUB '90
    :PRINT TAB(K);":   PROCESS   :"
    :GOSUB '90
    :PRINT TAB(K);":............:"
    :X0=1
    :Z0=1
3580 IF E=0  THEN K=1
    :GOSUB '90
    :PRINT TAB(K);":-------------:"
    :IF Z0=1  THEN GOSUB '93
    :IF Z0=8  THEN Z0=0
    :IF Z0[]1  THEN GOSUB '90
3590 Z0=Z0+1
    :$TRAN(B2$,"..............................")
    :$TRAN(A2$,"..............................")
3600 IF A2$="INPUT" OR A2$="KEYIN" OR A2$="LINPUT"  THEN PRINT TAB(K);":
     I/O  ";A2$;TAB(K+14);":  ";B2$
3610 IF STR(A2$,1,2)=STR(B2$,1,2)  THEN B2$=" "
3620 IF A2$[]"INPUT" AND A2$[]"KEYIN" AND A2$[]"LINPUT"  THEN PRINT TAB(K
    );":  ";STR(A$,B,E1-B);TAB(K+14);":  ";A2$;" ";B2$
    :RETURN
```

PROGRAM LISTING

```
3630 REM %

    SUBROUTINE (DEFFN'8) ........

    .....................................
3640 REM print down arrow-line
3650 DEFFN'8
    :GOSUB '90
    :PRINT TAB(K);"         !"
    :GOSUB '90
    :PRINT TAB(K);"         V"
    :GOSUB '90
    :PRINT TAB(K);"         !"
    :GOSUB '90
    :PRINT TAB(K);"         !"
    :RETURN

3660 REM %

    SUMROUTINE (DEFFN'4) ........

    .....................................
3670 REM print line and arrows, horizontal
3680 DEFFN '4
    :N,N1=0
    :CONVERT B2$  TO N1
    :ERRORN,N1=0
    :GOTO 3700

3690 CONVERT N$  TO N
    :ERRORN,N1=0
    :GOTO 3700

3700 GOSUB '90
    :PRINT TAB(K);"         !";
    :IF N1[N  THEN PRINT TAB(K+34);"/!"
    :ELSEPRINT " "
    :GOSUB '90
3710 PRINT TAB(K);"         V";
    :IF N1[N  THEN PRINT TAB(K+34);" !"
    :ELSEPRINT " "
    :GOSUB '90
    :PRINT TAB(K);"         !";
    :IF N1[N  THEN PRINT TAB(K+34);" !"
    :ELSEPRINT " "
    :GOSUB '90
3720 PRINT TAB(K);"         ----------------------]-----"
    :GOSUB '90
3730 IF N1]N  THEN PRINT TAB(K+34);" !"
    :ELSEPRINT " "
    :GOSUB '90
    :IF N1]N  THEN PRINT TAB(K+34);" !"
```

```
          :ELSEPRINT " "
          :GOSUB '90
          :IF N1 ]N  THEN PRINT TAB(K+34);" V"
          :ELSEPRINT " "
          :GOSUB '90
       3740 U=1
          :IF U6=0  THEN RETURN
          :U6,U=0
          :E=E-1
          :IF E=0  THEN PRINT "!"
          :RETURN

       3750 IF U[ ]1  THEN 3760
          :K=E*20
          :IF E=0  THEN K=0
          :U=0
       3760 IF E[1  THEN PRINT TAB(K);"        --[----- [ ? ]"
          :ELSEPRINT TAB(K);" "
       3770 GOSUB '90
          :PRINT TAB(K);"        V"
          :GOSUB '90
          :PRINT TAB(K);"        V"
          :GOSUB '90
          :PRINT TAB(K);"        !"
          :GOSUB '90
          :PRINT TAB(K);"        !"
          :RETURN

       3780 REM %

          SUBROUTINE (DEFFN'19) .......

          .................................
       3790 REM print illegal use of IF,THEN, / print module [==] unstructured
       3800 DEFFN'19
          :W3=W3+1
          :PRINT
          :PRINT "COMMENT,    CANNOT READ /IF...THEN/ IN STATEMENT (MAY BE TO L
          ONG TO PROCESS)"
          :PRINT "            OR CANNOT MATCH /FOR/ OR /NEXT/ WITH IT'S LOOP CO
          MPLEMENT"
          :PRINT  STR(A$,1)
       3810 RETURN

       3820 K=E*20
          :GOSUB '90
          :PRINT TAB(K);"        !"
          :GOSUB '90
          :PRINT TAB(K);"        !"
          :GOSUB '90
          :PRINT TAB(K);"     /    *"
          :GOSUB '90
```

PROGRAM LISTING

```
    :PRINT TAB(K);"    /      *"
3830 GOSUB '90
    :PRINT TAB(K);"   / MODULE *"
    :GOSUB '93
    :PRINT TAB(K);"  /    ";W1;TAB(K+13);"*"
    :GOSUB '90
    :PRINT TAB(K);" /            *"
    :GOSUB '8
3840 RETURN

3850 REM %

    SUBROUTINE (DEFFN '82) ......

    ..................................
3860 DEFFN'82
    :K=E*20
    :IF XO=1  THEN GOSUB '81
    :IF YO[]0  THEN 3890
    :FOR S3=1  TO 2
    :GOSUB '90
    :PRINT TAB(K);"        !"
    :NEXT S3
    :IF E=0  THEN K=1
    :GOSUB '90
3870 PRINT TAB(K);"  /===============/ ";N$
    :GOSUB '90
    :PRINT TAB(K);" /  -----------  /"
    :GOSUB '90
    :PRINT TAB(K);"(  ( PRINTING ( ("
    :GOSUB '90
    :PRINT TAB(K);" *  -----------  *"
    :GOSUB '90
    :PRINT TAB(K);"  *---------------*"
3880 YO=1
    :X=1
    :ZO=0
3890 IF E=0  THEN K=1
    :IF ZO=1  THEN GOSUB '93
    :IF ZO=8  THEN ZO=0
    :IF ZO[ ]1  THEN GOSUB '90
    :ZO=ZO+1
    :$TRAN(B2$,".............................")
    :IF B2$="RINT.."  THEN B2$="1 BLANK LINE"
    :IF MOD(X,2)=0  THEN 3910
3900 PRINT TAB(K);"    )--------------)"
    :GOSUB '90
    :PRINT TAB(K);"  /               /"
    :GOSUB '90
    :IF A2$="PRINT"  THEN PRINT TAB(K);" /    PRINT      /  ";B2$
    :ELSEPRINT TAB(K);" /  PRINTUSING   /  ";B2$
    :GOTO 3920
```

PROGRAM LISTING

```
3910 PRINT TAB(K);"(---------------("
    :GOSUB '90
    :PRINT TAB(K);" *                *"
    :GOSUB '90
    :IF A2$="PRINT"  THEN PRINT TAB(K);"  *    PRINT      * ";B2$
    :ELSEPRINT TAB(K);"  *   PRINTUSING  * ";B2$
3920 X=X+1
    :RETURN

3930 REM %

    SUBROUTINE (DEFFN '83) ......

    ..............................
3940 DEFFN '83
    :K=E*20
    :YO=0
    :IF E=0  THEN K=1
    :GOSUB '90
    :IF MOD(X,2)=0  THEN PRINT TAB(K);"/==============/"
    :ELSEPRINT TAB(K);"   ================="
    :IF E=0  THEN K=0
    :GOSUB '8
    :RETURN

3950 REM %

    SUBROUTINE (DEFFN '66) ......

    ................................
3960 DEFFN '66
    :IF U6=0  THEN RETURN
    :IF U6=2 AND STR(A$,1,J-1)=N$  THEN RETURN
    :U6=0
    :IF XO=1  THEN GOSUB '81
    :IF YO=1  THEN GOSUB '83
    :E=E-1
    :K=E*20
    :IF E=0  THEN K=1
    :PRINT TAB(K);"       !----------[--------!"
    :RETURN

9500 REM %

   GENERAL SUBROUTINE XXXXXXXXXX

    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    :PRINT HEX(030E);"YOU HAVE LOADED 'DEFFNO' THE GENERAL SUBROUTINES";
    HEX(0A)
    :LIST
    :END
```

PROGRAM LISTING

```
9530 DEFFN 'O "SCRATCHT";HEX(22);"BASCHART";HEX(22);":SAVET$ ()";HEX(22)
    ;"BASCHART";HEX(2200)
9570 DEFFN '31
    :GOTO 1430
```